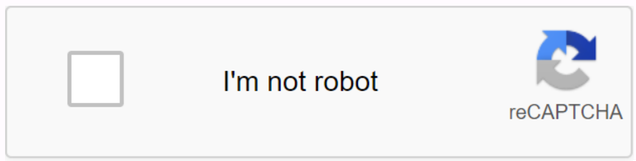

Continue

Er diagram solved examples pdf

The Entity and Relationship Model (ER Model) describes the structure of the database using a diagram, which is known as the Entity Relationship Diagram (ER Diagram). An ER model is a design or draft database that can later be deployed as a database. The main components of the E-R model are: an entity set and a set of relationships. What is an Entity Relationship Diagram (ER Diagram)? The e-diagram shows the relationship between entities. An entity set is a group of similar entities, and these entities can have attributes. As these dbms are, an entity is a table or table attribute in a database, so that by displaying the relationships between tables and their attributes, the ER diagram shows the complete logical structure of the database. Let's look at a simple ER diagram to understand this concept. Simple e-diagram: In the following diagram we have two Student and College entities and their relationship. The relationship between a student and a faculty is much one as a faculty can have many students, however a student cannot study at multiple faculties at the same time. A student entity has attributes such as Stu_Id, Stu_Name & Stu_Addr, and a College entity has attributes such as Col_ID & Col_Name. Here are the geometric shapes and their meaning in the E-R diagram. We will discuss these terms in detail in the next section (Components of the ER diagram) of this guide, so do not worry too much about these conditions now, just foresee them once. Rectangle: Represents entity sets. Ellipses: Diamond Attributes: Relationship Set Lines: They link attributes of entity sets and entity sets to double relationship set ellipses: Multiple attributes of a intermittent ellipse: Derived attributes Double Rectangles: Weak Entity Sets Double Lines: Total Entity Participation in Relationships Set By ER Diagram Components As Shown in Diagram Above, The ER diagram has three main components: 1. Attribute 3. Relationship 1. The Entity Entity is an object or component of a data. The entity is represented as a rectangle in an ER diagram. For example: In the following ER diagram, we have two Student and College entities and these two entities have many to one relationships because many students study at one faculty. We'll read more about relationships later, focus on entities for now. Weak Entity: An entity that cannot uniquely identify itself with its own attributes and relies on a relationship with another entity is called a weak entity. A weak entity represents a double rectangle. For example – a bank account cannot be uniquely identified without knowing the bank to which the account belongs, so the bank account is a weak entity. 2. Attribute attribute describes the property of the entity. The attribute is represented as Oval in the ER diagram. There are four types of attributes: 1. Key attribute 2. Composite Attribute 3. Multiple Attribute 4. Derived Attribute 1. Key Attribute: A key attribute can uniquely identify an entity from an entity set. For example, the number of studentrol identify students from the student set. The key attribute represents the oval same as other attributes, however the key attribute text is underlined. 2. Composite Attribute: An attribute that is a combination of other attributes is known as a composite attribute. For example, in a student entity, a student address is a composite attribute because the address consists of other attributes such as pin code, status, country. 3. Multiple Attribute: An attribute that can hold multiple values is known as a multiple attribute. It is presented with double ovals in the ER diagram. For example - a person can have more than one phone number so the phone number attribute is multifaceted. 4. Derived Attribute: A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by a dodged oval in an ER diagram. For example – A person's age is a derived attribute because it changes over time and can be extracted from another attribute (Date of Birth). E-R diagram with multiple and derived attributes: 3. Relationship A relationship is represented by a diamond shape in an ER diagram, shows the relationship between entities. There are four types of relationships: 1. one to one 2. One to many 3. Many to one 4. Much to many 1, 1,000 people. One-on-one relationship When one entity case is linked to one example of another entity, then it is called one-on-one relationship. For example, a person has only one passport, and a passport is given to one person. 2. One to many relationships When one entity case is linked to multiple cases of another entity, then it is called one on many relationships. For example – a customer can place a lot of orders, but many customers can't place an order. 3. Many to one relationship When more than one entity case is related to one example of another entity, then many are called a single relationship. For example – many students can study at one faculty, but a student cannot study at many faculties at the same time. 4. Many relationships When more than one entity case is linked to multiple cases of another entity, then many are called many relationships. For example, they can be assigned to many projects, and the project can be assigned to many students. The total participation of an entity determined by the total participation of an entity set represents that each entity in the entity set must have at least one relationship in the relationship set. For example: In the diagram below, each faculty must have at least one affiliated student. THE ENTITY RELATIONAL (ER) MODEL is a diagram of a high-level conceptual data model. ER modeling helps you systematically analyze data requests to produce a well-designed database. The model of entities and relationships represents the real-world subjects and the relationship between them. It is considered best practice to complete ER modeling before implementing your database. ER modeling helps you analyze data requirements to create a well-designed database. Thus, it is considered best practice to complete ER modeling before implementing your database. In this guide, you will learn- The history of ER modelsER diagrams are a visual tool that is useful for presenting ER models. Peter Chen proposed it in 1971. He aimed to use the ER model as a conceptual modeling approach. THE ENTITY AND RELATIONSHIP (ERD) DIAGRAM SHOWS the entity relationships set up in the database. In other words, we can say that ER diagrams help you explain the logical structure of databases. At first glance, the ER diagram looks very similar to a flowchart. However, the ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of the ER Diagram is to present the framework infrastructure of the entity. Sample ER Diagram Facts about ER Diagram Model: ER model allows you to draw a database design It is easy to use graphical data modeling tool Widely used in database design It is a GUI view of the logical structure of the database Helps you identify the entities that exist in the system and the relationships between these entities Why use ER diagrams? Here are the main reasons to use the ER Diagram helps you define terms related to modeling entity relationships Provide an overview of how all your tables should be linked, Which fields will be on each table Helps you describe entities, attributes, relationships ER diagrams are transferable to relational tables that allow you to quickly build databases ER diagrams can be used by database designers as a blueprint for the implementation of data in certain software applications to better understand the information that will be contained In the database with the help of the ERD ERP diagram, it allows you to communicate with the logical database structure to users of the ER DiagramThis model component based on three basic concepts: Attributes RelationshipsExplaigned For example, in a university database, we may have entities for students, courses and lecturers. A student entity can have attributes such as Rollno, Name, and DeptID. Maybe they have something to do with courses and lecturers. WHAT IS AN ENTITY? A thing in the real world either alive or non-alive that is easily recognizable and unrecognizable. This is all in the company that will be represented in our database. It can be a physical thing or simply a fact about an enterprise or an event that happens in the real world. An entity can be a location, person, object, event, or concept, that stores data in a database. Entity characteristics must have an attribute and a unique key. Each entity appears to be of some attributes representing that entity. Examples of entities: Person: Employee, Student, Patient Location: Shop, Building: Machine, Product and Car Event: Sales, Registration, Renewal Concept: Note an entity course set:Student Entity Set is a group of similar entity types. It can contain entities with an attribute that share similar values. Entities are their properties, also called attributes. All attributes have their own separate values. For example, a student entity can have a name, age, class, as attributes. Entity example: A university may have some departments. All these departments employ various lecturers and offer several programs. Some courses make up each program. Students enroll in a particular program and enroll in different courses. The lecturer from a certain department attends each course, and each lecturer teaches a different group of students. RelationshipRelationship is nothing more than an association between two or more entities. For example, Tom works in the chemistry department. Entities participate in relationships. We can often identify relationships with verbs or verb phrases. For example: You are attending this lecture I teach only local entities, we can classify relationships to types of relationships: Student attends lecture Lecturer gives lecture. A weak entityA weak entity is a type of entity that does not have its key attribute. It can be identified uniquely taking into account the primary key of another entity. For this, weak entity sets must participate. For example, Trans No is discriminatory within the ATM transaction group. Let's learn more about a weak entity by comparing it to a strong entity a strong entity set by a weak entity set strong entity A set of strong entities always has a primary key. There are not enough attributes to build the primary key. It is represented by a rectangle symbol. It is represented by a double rectangle. Contains the primary key represented by the underline symbol. Contains a partial key represented by a intermittent underline symbol. A member of a strong entity is called the dominant entity. Member of a weak entity set up as a child entity. The primary key is one of its attributes that helps identify its member. In a set of weak entities, this is a combination of the primary key and the partial key of a strong entity set. The ER e-diagram shows the relationship between two strong entities displayed using a diamond symbol. The relationship between one strong and weak entity is shown using a double diamond symbol. The connection line of a strong entity set to a relationship is unique. A line that links a weak entity set up to identify relationships is twofold. AttributesIt is a property with one valuable valuable entity type or relationship type. For example, a lecture can have attributes: time, date, duration, location, etc. The attribute is represented by the Ellipse Types of Attributes Description Simple attribute Simple attributes cannot be further shared. For example, the contact number of students. It is also called atomic value. Composite Attribute Can Be Broken composite attribute. For example, a student's full name can be further divided into first name, second name, and last name. Derived attribute This attribute type is not included in the physical database. However, their values derive from other attributes present in the database. For example, age should not be stored directly. Instead, this should be derived from that employee's AGE. Multiple Attributes Multiple attributes can have more than one value. For example, a student may have more than one mobile phone number, email address, etc. CardinalityDefined numeric attributes of relationships between two entities or entity sets. Different types of cardinal relationships are: One-to-one one-to-many relationships Relationships can be many-to-many 1.One-to-one relationships: One entity from an entity set up by X can be linked to most of one entity entity set up by Y and vice versa. Example: One student can apply for a number of courses. However, all these courses have one line back to that one student. 2.One-on-many: One entity from the X Set entity may be associated with multiple entity entities set up by Y, but an entity from an entity set by Y can be linked to at least one entity. For example, one class consists of multiple students. 3. Many to one more than one entity from the X-Series entity may be associated with most of one entity entity set up by Y. However, an entity from an entity set up by Y may or may not be associated with more than one entity from the X-set entity. Many to many: One entity from X can be associated with more than one entity from Y and vice versa. For example, Students as a group are affiliated with multiple faculty members, and faculty members can be associated with multiple students. ER- Diagram NotationsER- Diagram is a visual representation of data describing how data is interconnected. Rectangles: This symbol represents entity types of Ellipses : Symbol represents diamond attributes: This symbol represents line relationship types: Linking entity attributes and entity types to other types of Primary Key relationships: attributes are underlined Double Ellipses: They represent the multifunctional Steps attributes Steps to create ERDFollowing are steps to create ERDFollowing. We teach them with an example: At university, a student enrolls in courses. The student must be assigned at least one or more courses. Each course is taught by one professor. To maintain the quality of teaching, the professor can deliver only one course Step 1) Identification of entitiesIma we have three entities Step 2) Identification of relationshipsIma we have the following two relationships Student is assigned courseProfessor brings course Step 3) Identification of cardinalityFor them statement about the problem we know that, student can be given multiple coursesA Professor can deliver only one course Step 4) Identify attributes You need to study files , forms , reports, data currently by the attribute identification organization. You can also interview different stakeholders to identify entities. Initially, it is important to identify attributes without mapping a specific entity. After you have a list of attributes, you must map them to the identified entities. Ensure that the attribute is paired with exactly one entity. If you think the attribute should belong to more than one entity, use the modifier to make it unique. Once the mapping is done, identify the primary keys. If a unique key isn't easily accessible, create one. Entity primary key attribute student Student_ID StudentName professor Employee_ID ProfessorName course Course_ID CourseName for the entity course, attributes can be duration, points, tasks, etc. For ease, we considered only one attribute. Step 5) Create an ERDA more modern view of the best practices of ERD diagrams for developing effective ER diagrams Eliminate all redundant entities or relationships You need to make sure that all your entities and relationships are properly labeled There may be different valid approaches to the ER e-diagram. You need to make sure that the ER diagram supports all the data you need to store You should make sure that each entity appears only once in the ER diagram Specify each relationship, entity and attributes are represented in your diagram Never link relationships with each other You should use colors to highlight important parts of the ER diagram Summary Er model is a diagram of a high-level ER diagram of a data model are a visual tool that is useful to represent the diagram of the relationship of the ER model Entity shows the relationships of the entity Stored in ER database diagrams that help you define terms related to the ER model relationship model model, the model is based on three basic concepts: Entities, Attributes, and Relationships An Entity can be a location, person, object, event, or concept, which stores data in a Relationship database is nothing more than a connection between two or more entities A weak entity is a type of entity that does not have its key attribute It is a single-value property of any entity or relationship type Helps you define numeric attributes of relationships between two entities or entity sets ER- Diagram is a visual representation of data that describes how data is related to each other While drawing an ER diagram you need to make sure that all your entities and relationships are correctly selected. Page 2RELATIONAL ALGEBRA is the widely used language of a procedural query. It collects relationship cases as input and gives relationship occurrences as an exit. Use different operations to perform this action. The result of these operations is a new relationship, which can be formed from one or more input relationships. In this tutorial, you will learn: Basic SQL relational algebra operationRelational Algebra devided in Group Unary Relational OperationsSELECT (symbol: σ)PROJECT (symbol: π)RENAMING (symbol: η)Relational algebra operations from the set theoretical intersection ( ), DIFFERENCE (-) CARTESIAN PRODUCT (x)Binary relational operationsNew them in detail with solutions: SELECT (σ)Operation SELECT is used to select the subset of the tuples according to a specific selection state. Sigma (σ)Symbol indicates it. Used as a term to select tuples that meet the selection condition. The selected operator selects tuples that meet a specific predicate. tp(r) σ is predicate r standing for a relationship that is the name of table p is pre-position logic Example 1 σ topic = Database (Tutorials) Output - Selects tuples from Tutorials where topic = 'Database'. Example 2 σ topic = Database and author = guru99(Tutorials) Output - Selects tuples from Tutorial where the theme is 'Database' and 'author' is guru99. Example 3 σ Sales > 50000 (Customers) Output - Select a tuples from customers where sales are greater than 50000 Projections (π)Projection eliminates all attributes of the input relationship, but those specified in the list of projections. The projection method defines a relationship that contains a vertical subset of Relationships. This helps extract the values of the specified attributes to remove duplicate values. (pi) the symbol is used to select attributes from a relationship. This operator helps you keep certain columns from relationships and discards other columns. Projection Example: Consider the following CustomerID CustomerName Status 1 Google Active 2 Amazon Active 3 Apple Inactive 4 Alibaba Active Here table, a customername projection and status will give Π CustomerName, CustomerName Status (Customers) Google Active Amazon Active Apple Inactive Alibaba Active Rename (η)Renaming is an unautical operation used to rename relationship attributes. η (a/b)R will rename attribute b of the relationship to a. The Union Operation (η)UNION symbolizes the ∪ symbol. Includes all tuples that are in tables A or in B. Also eliminates duplicate tuples. Thus, to set the UNION B set would be expressed as: Result <- A ∪ B For a union operation to be valid, the following conditions must be maintained - R and S must be the same number of attributes. Domain attributables must be compatible. Duplicate tuples should be removed automatically. Example Consider the following tables. Table B column 1 column 2 column 1 column 2 1 1 1 1 1 1 1 2 1 3 ∪ B gives the table column ∪ B 1 column 2 1 1 1 1 2 1 3 Set the difference (-)- The symbol indicates it. Result A - B, is a relationship that includes all tuples that are in A but not in B. Attribute name A must match attribute name in B. Two-form relationships A and B should be compatible or compatible with the Union. A relationship consisting of tuples that are in relation to A but not in B.Example A-B Table A - B column 1 column 2 1 2 Intersection is defined by the symbol of the ∩ ∩ B Defines a relationship consisting of a set of all tuples that are in A and B. However, A and B must be compatible with the union. Visual Definition intersectionExample: A ∩ B Table A ∩ B column 1 column 2 1 1 Carthesian product in DBMS is an operation used to merge columns from two relationships. In general, a Cardiasian product is never a significant operation when working alone. However, it becomes meaningful when followed by other operations. It's also called Cross Product or Cross Join. Example – Cartesian product σ column 2 = '1' (A X B) Output – The above example shows all rows from A-B relationships whose column 2 has a value of 1 σ column 2 = '1' (A X B) column 1 column 2 1 1 1 Join operationsJoin work is basically a cartesian product followed by a selection criterion. Join the operation ⋈. JOIN also allows you to join differently connected tuples from different relationships. Types JOIN:Different forms of association operation are: Inner Joins: Theta joinEQUI join Natural joinOuter join: Left Outer JoinRight Outer JoinFull Outer JoinInner Join: In a inner join, only those tuples that meet the match criteria are included, while the rest are excluded. Let's try different types of internal merged: Theta Join:General case of JOIN operation is called Theta join. It is marked with a θ A ⋈θ BTheta can use any conditions in the selection criteria. For example: column ⋈ A.column 2 > B.column 2 (B) A ⋈ A.column 2 > B.column 2 (B) column 1 column 2 1 2 EQUI join:When a theta join uses only an equi join condition. For example: column ⋈ A.column 2 = B.column 2 (B) A ⋈ A.column 2 = B.column 2 (B) column 1 column 2 1 EQUI join is the most difficult operation for effective implementation using SQL in RDBMS and one of the reasons why RDBMS has essential performance issues. NATURAL JOIN (⋈)A natural port can only be exported if there is a common attribute (column) between relationships. The name and type of attribute must be the same. Example Consider the following two tables C ⋈ D C ⋈ D Num Square Cube 2 4 4 3 9 27 OUTER JOIN In the outer join, in addition to tuples that meet the match criteria, we also include some or all of the tuples that do not match the criteria. Left outer join(A B)In the left outer port, the operation allows you to keep all the tuple in the left relationship. However, if there is no corresponding tuple in the right relationship, then the attributes of the right relationship in the association result are filled with null values. Consider the following 2 tables Number Square 2 4 3 9 4 16 A B A ⋈ B Num Square Cube 2 4 4 3 9 9 4 16 - Right External Association: (A B) In the right external association, the operation allows you to maintain all the tuple in the right relationship. However, if there is no corresponding tuple in the left relationship, then the left relationship attributes in the association result are filled with null values. A B A ⋈ B Num Cube Square 2 8 4 3 18 9 5 75 - Full Outer Join: (A B)In full external association, all tuples from both relationships regardless of the appropriate situation. A B A ⋈ B Num Cube Square 2 4 8 3 9 18 4 16 - 5 - 75 SummaryOperation(Symbols) The select purpose (σ) Operation SELECT is used to select a sub-group of tuples according to a specific projection selection condition (π) The projection eliminates all attributes of the input relationship, but those listed in the list of projections. The Union (∪) symbolizes the symbol. Includes all tuples contained in tables A or in B. Set Difference (-) - Symbol indicates it. Result A - B, is a relationship that includes all tuples that are in A, but not in B. Intersection (∩) Intersection defines a relationship consisting of a set of all tuples located in both A and B. Cartesian Product (X) Cartesian operation is useful to

connect the pillars from the two relationships. Inner Join Inner join, includes only those tuples that meet the match criteria. Theta Join (θ) The general case of OPERATION JOIN is called Theta join. Is marked with θ. EQUI Join When theta you join uses only equivalence state, it becomes equi join. Natural Port (⋈) A natural port can only be exported if there is a common attribute (column) between relationships. Outer Join In outer join, complete with tuples that meet match criteria. Left Outer Join() In the left external port, the operation allows you to keep all the tuple in the left relationship. Right Outer Join() In the right outer join, the work allows you to keep all the tuple in the right relationship. Full Outer Join() In full outer join, all tuples from both relationships are included in the result regardless of the appropriate condition. Page 3A Database Transaction is a logical processing unit in DBMS that involves one or more database access operations. In short, database transactions represent real-world events of any company. All types of database access operations held between transaction start and end statements are considered to be one logical transaction in DBMS. During a transaction, the database is inconsistent. Only after the database has been submitted does the state change from one consistent state to another. In this guide, you will learn: Transaction Transaction Facts is a program unit whose execution may or may not change the contents of the database. The transaction concept in DBMS is executed as a single unit. If database operations do not update the database, but only retrieve data, this type of transaction is called a read-only transaction. A successful transaction can change a database from one consistent country to another DBMS transactions must be atomic, consistent, isolated and durable If the database was in an inconsistent state before the transaction, it would remain in an inconsistent state after the transaction. Why do you need consent in transactions? A database is a shared resource that is accessed. It is used by many users and processes simultaneously. For example, the banking system, rail and air traffic booking systems, supermarket and cash register inventory, etc. Failure to manage concurrent access can create problems such as: Hardware failure and crash At the same time executing the same transaction, downtime, or slow transaction performance Different transaction concept states in DBMS are listed below: Types of government transactions Active state Transaction enters active state when execution process begins. During this state, reading or writing operations can be performed. The partially submitted Transaction goes into a partially committed state after the transaction is completed. Surrendered State When the transaction is submitted to the state, it has already successfully completed the execution. Moreover, all its changes are recorded in the database permanently. A failed country Transaction is considered unsuccessful when any of the checks fail or if the transaction is terminated while it is in an active state. An aborted transaction state reaches an aborted state when certain transactions leaving the system cannot be restarted. State transition diagram for database transaction Let's study a state transition diagram that highlights how a transaction moves between these different states. After the transaction specifies execution, it becomes active. It can publish a READ or WRITE operation. Once reading and WRITING operations are completed, the transactions become partially submitted. Then, some recovery protocols must ensure that the system failure will not result in the inability to permanently record changes in the transaction. If this check is successful, the transaction is committed and enters the submitted state. If the check failed, the transaction goes into a failed state. If the transaction is terminated while it is in an active state, it goes into a failed state. The transaction should be returned to reverse the effect of its writing operations in the database. The aborted state refers to a system abandonment transaction. ACID Properties are used to maintain database integrity during transaction processing. ACID in DBMS indicates atomicity, consistency, insulation and durability. Atomicity: A transaction is one unit of work. Either execute it completely or you don't execute it at all. It can't be a partial execution. Consistency: After executing a transaction, it should move from one consistent state to another. Isolation: The transaction should be carried out separately from other transactions (without locks). While executing transactions simultaneously, the results of intermediate transactions from simultaneously executed transactions should not be available to each other. (Level 0.1,2,3) Durability: · · Once the transaction has been completed successfully, changes to the database should continue. Even in case of system failures. Acid Real Estate in DBMS with example:Below is an example of ACID assets in DBMS: Transaction 1: Begin X=X+50, Y = Y-50 END Transaction 2: Start X = 1.1 *X, Y = 1.1 *Y END Transaction 1 transfers $50 off X to account Y. Transaction 2 credits each account with an interest payment of 10%. If both transactions are submitted together, there is no guarantee that transaction 1 will take place before transaction 2 or vice versa. Regardless of the order, the result must be as if the transactions are conducted serially one after the other. Types of transactions based on scopes that are not distributed in relation to distributed transactions Transactions Transactions timed online vs. series Based on actions A two-way model of a restricted action based on the structure of a straight or simple transaction: It consists of a series of primitive operations performed between the beginning and end of operations. Nested Transactions: A transaction that contains other transactions. Workflow What is a schedule? A schedule is the process of creating one group of multiple parallel transactions and executing them one by one. It should preserve the order in which the instructions appear in each transaction. If two transactions are executed simultaneously, the result of one transaction may affect the achievement of another transaction. Example Initial Product Quantity is 10 Transaction 1: Update product quantity to 50 Transactions 2: Read the product quantity If transaction 2 is made before transaction 1, outdated product quantity information will be read. Therefore, schedules are required. Parallel execution in the database is inevitable. But parallel execution is allowed when there is an equivalence of relationships between simultaneous execution of transactions. This equivalence is of 3 species. EQUIVALENCE OF RESULTS:If two schedules display the same result after execution, this is called the result equivalent schedule. They can offer the same result for a value and different results for another set of values. For example, one transaction updates the product quantity, while the other updates customer information. View Equivalence equivalence view occurs when a transaction in both schedules performs a similar action. For example, one transaction inserts product details into the product table, while another transaction inserts product details into the archive table. The transaction is the same, but the tables are different. CONFLICT Equivalence In this case, two transactions update/view the same data set. There is a conflict between transactions because the order of execution will affect the result. What is serieslibility? Serialization is a process of searching for a simultaneous schedule that is equal to a serial schedule in which an ae transaction executes one after the other. Depending on the type of schedule, there are two types of serialization: Summary: Transaction management is a logical processing unit in DBMS that implies one or more database access operations It is a transaction of a program unit whose execution may or may not change the contents of the database. Failure to manage con-time access can create problems such as hardware failure and crashing. Active, partially committed, committed, failed and terminated country of transactions. Full form of ACID Properties in DBMS are Atomicity, Consistency, Isolation, and Durability Three types of DBMS transactions are Base in application areas, Action and Structure. A schedule is the process of creating one group of multiple parallel transactions and executing them one by one. Serialability is a process of searching for a concurrent schedule whose output is equal to a serial schedule in which transactions are executed one after the other. Page 4Konkurecy control is a process in DBMS to manage simultaneous operations without conflicting with each other. Simultaneous access is very simple if all users are just reading the data. There's no way they're interfering with each other. Although for any practical database, it will have a mixture of reading and WRITING operations and therefore consent is a challenge. Concurrent control is used to resolve such conflicts that occur mainly with a multi-user system. Helps ensure that transactions in the database are done simultaneously without violating the integrity of the data of the relevant databases. Therefore, concurrent control is the most important element for the proper functioning of a system in which two or more database transactions are simultaneously executed requiring access to the same data. In this guide, you'll learn Concurrency Potential Issues Here, some of the issues you're likely to face while using the configuration control method: Lost updates occur when multiple transactions select the same line and updates the line based on the values of the selected Unsealed Dependency Issues occur when another transaction selects a line that is updated with another transaction (dirty reading) Fixed Read occurs when another transaction tries to access the same row several times. and reads different data each time. The wrong problem summary occurs when one transaction takes a summary of the value of all cases of a repeat data item, and the other transaction updates several cases of that particular data item. In this situation, the summary obtained does not reflect the exact result. Why use the concurrency method? The reasons for using the concurrent control method is DBMS: Apply isolation through mutual exclusion between conflicting transactions In order to resolve issues of reading and writing conflicts in order to preserve the consistency of the database through the constant preservation of execution obstructions The system needs to control the interaction between concurrent transactions. This control is achieved by contest control programs. Concurrent control helps ensure a serial exampleAssume that two people who go to electronic kiosks at the same time buy a movie ticket for the same movie and the same time of the show. However, there is only one place left for a film performance in that theatre. Without concurrent control, it is possible that both filmmakers will eventually buy a ticket. However, the method of controlling concurrently does not allow this to happen. film buffs can still access information written in the film seating database. But concurrent control provides only a ticket to the customer who first completed the transaction process. Serenity Control ProtocolsDifferent consent control protocols offer different benefits between the amount of consent they allow and the amount of overheads they impose. Lock-Based Protocols Two PhaseTimestamp-based Protocols Validation-Based Protocols Lock-based ProtocolsA lock is a data variable that is linked to a data item. This lock indicates operations that can be performed on a data item. Locks help synchronize access to database items through contestacit transactions. All lockout requests are addressed to the consent control manager. Transactions only resume after a lock request is granted. Binary locks: A binary lock on a data item can lock or unlock states. Shared/exclusive: This type of locking mechanism separates the locks based on their use. If a lock is obtained on a data item to perform a write operation, this is called an exclusive lock. 1. Common lock (S): The common lock is also called a read-only lock. With a shared lock, a data item can be shared between transactions. This is because you will never have permission to update data item data. For example, consider a case where two transactions read the balance of a person's account. The database will allow them to read by setting a common lock. However, if another transaction wants to update the balance of that account, the shared lock prevents it until the reading process is complete. 2. Exclusive lock (X): With the Exclusive Lock, a data item can be read and written. This is exclusive and cannot be maintained simultaneously on the same data item. X-lock is required using lock-x instructions. Transactions can unlock a data item after the write operation is complete. For example, when a transaction needs to update a person's account balance. You can allow this transaction by putting an X lock on it. Therefore, when another transaction wants to read or write, an exclusive lock prevents this operation. 3. Simplified lock protocol This type of lock-based protocol allows transactions to obtain the locking of each object before starting work. Transactions can unlock a data item after the write operation is complete. 4. Pre-claiming Locking Pre-claiming lock protocol helps evaluate operations and create a list of required data items that are required to initiate the execution process. In a situation where all locks are approved, the transaction is executed. After that, all locks are released when all his surgeries are done. Starvation of hunger is a situation when a transaction needs to wait indefinitely to acquire a lock. The following are the reasons for starvation: When the waiting scheme for locked items is not properly managed the same transaction was selected as a victim multipleDeadlock Stalemate refers to a particular situation two or more processes are waiting for each other to release a resource or more than two processes are waiting for a resource in a circular chain. Two phase lock protocols (2PL) protocolTwo-phase locking which is also known as the 2PL protocol. It's also called P2L. In this type of lock protocol, a transaction should acquire a lock after it releases one of its locks. This locking protocol divides the execution phase of the transaction into three different parts. In the first stage, when the transaction begins to execute, permission is required for the locks it needs. The second part is where the transaction gets all the locks. When a transaction publishes its first lock, the third stage begins. At this third stage, the transaction cannot require new locks. Instead, it releases only acquired locks. The two-stage lock protocol allows each transaction to make a two-step lock or unlock request: Growth phase: At this stage, the transaction can get locks but cannot free the locks. Downsizing phase: At this stage the transaction can release locks, but not get a new lockIt is true that the 2PL protocol offers serialalization. However, this does not ensure that setbacks do not occur. In the diagram above, you can see that local and global downtime detectors are looking for downtime and resolving them by continuing transactions to their initial states. The strict two-phase locking systemStrict-Two phase locking system is almost similar to 2PL. The only difference is that Strict-2PL never releases the lock after use. Holds all locks to the point of rest and releases all locks at one time when the process is complete. Centralized 2PLIn Centralized 2 PL, one page is responsible for the lock management process. There's only one lock manager for the entire DBMS. The primary copy of the 2PLPrimary copying 2PL mechanism, many lock managers are distributed to different sites. After that, a specific lock manager is responsible for managing the lock for a set of data items. When the primary copy is updated, change is propagated to slaves. Distributed 2PLU to this type of two-file locking mechanism, Lock Managers is distributed to all sites. They are responsible for managing the data locks at the site. If the data is not replicated, this is equal to the primary copy of 2PL. The communication costs of distributed 2PL are quite higher than the primary copy of the Timestamp-based 2PL Protocols The health-time algorithm uses timestamp to serialize the execution of concurmental transactions. This protocol ensures that all conflicting read and write operations are performed in the order of timetamp. The protocol uses system time or a logical number as a time game. An older transaction always takes precedence in this method. Use system time to determine the timestamp of a transaction. This is the most commonly used concurrent protocol. Lock-based protocols help you manage the order between transactions when they will execute. Execute. protocols manage conflicts as soon as an operation is created. Example: Suppose there are transactions T1, T2, and T3. T1 entered the system at the time when 0010 T2 entered the system at 0020 T3 has entered the system on 0030 Priority will be given to transaction T1, then transaction T2 and finally Transaction T3. Advantages: Schedules are serially similar to 2PL protocolsAny wait for a transaction, which eliminates the possibility of downtime! Disadvantages: Starvation is possible if the same transaction restarts and continuously interrupts the characteristics of the good consent protocol Ideal concurrent control DBMS mechanism has the following goals: It must be resistant to site failures and communication. Allows you to execute transactions in parallel to achieve maximum contestaly. Its storage mechanisms and computer methods should be modest to reduce costs. It must implement certain limitations in the structure of the atomic actions of transactions. Summary Concurrent Control is a process in DBMS to manage simultaneous operations without conflicting with each other. Lost updates, dirty reading, non-repeat reading, and an incorrect summary question are problems they face due to a lack of consent control. Lock-Based, Two-Phase, Timestamp-Based, Validation-Based are types of protocols for conpetual handling The lock could be shared (S) or Exclusive (X) A two-day lock protocol that is also known as a 2PL protocol need transaction should acquire a lock after it publishes one of its locks. It has 2 stages growing and decreasing. A time-based algorithm uses a man-made card to serialize the execution of concurral transactions. The protocol uses system time or a logical number as a time game. Page 5KEYS in DBMS is an attribute or set of attributes that helps you identify a row (tuple) in a relationship (table). They allow you to find a relationship between the two tables. Keys help you uniquely identify a row in a table by combining one or more columns in that table. The key is also useful for finding a unique record or row from a table. A database key is also useful for finding a unique record or row from a table. Example: Employee ID FirstName LastName 11 Andrew Johnson 22 Tom Wood 33 Alex Hale In the example above, an employee ID is the primary key because it uniquely identifies employee records. In this table, no other employee can have the same employee ID. In this guide, you will learn: Why do we need a key? Here are some reasons to use sql key in the DBMS system. Keys help you identify any row of data in the table. In the actual application, the table could contain thousands of records. Moreover, records could be duplicated. Keys ensure that you can uniquely identify a table record despite these challenges. Allows you to establish relationships between and identify relationships between Help tables to conduct identity and integrity in the relationship. In DBMS there are mostly seven different types of Key has different functionalities: Super Key - A super key is a group of individual or multiple keys that identifies rows in a table. Primary Key - is a column or group of columns in a table that uniquely identifies each row in that table. Candidate Key - is a set of attributes that uniquely identify the tuples in the table. Candidate Key is a super key without repeated attributes. Alternate Key - is a column or group of columns in a table that uniquely identifies each row in that table. Foreign key - is a column that creates a relationship between two tables. The purpose of foreign keys is to maintain data integrity and enable navigation between two different entity cases. Key Merge - has two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique in itself within the database. Composite Key - An artificial key that aims to uniquely identify each record is called a surrogate key. These key types are unique because they are created when you do not have any natural primary key. Replacement Key - An artificial key that aims to uniquely identify each record is called a surrogate key. These key types are unique because they are created when you do not have any natural primary key. What's a Super Key? Superkey is a group of individual or multiple keys that identifies rows in a table. A super key may have additional attributes that are not required for unique identification. Example: EmpSSN EmpNum Empname 9812345098 AB05 Displayed 9876512345 AB06 Roslyn 199937890 AB07 James In the example above, EmpSSN and EmpNum name are superkeys. A PRIMARY KEY is a column or group of columns in a table that uniquely identifies each row in that table. The primary key cannot be a duplicate, which means that the same value cannot appear more than once in a table. A table cannot have more than one primary key. Rules for defining primary key: Two rows cannot have the same primary key value Must each row have a primary key value. The primary key field cannot be void. The value in the primary key column can never be modified or updated if either foreign key refers to that primary key. Example: In the following example, <code>StudID</code> is the primary key. StudID Roll No First Name LastName Email 1 11 Tom Price This email address is being protected from spambots. You need JavaScript enabled for viewing. 2 12 Nick Wright This email address is being protected from spambots. You need JavaScript enabled for viewing. 3 13 Days Natan This email address is being protected from spambots. You need JavaScript enabled for viewing. ALTERNATE KEYS is a column or group of columns in a table that uniquely identifies each row in that table. A table can have multiple choices for a primary key, but only one can be set as the primary key. All keys other than the primary key are called Alternative Key. Example: In this table, StudID, But Email is qualified to become the primary key. But since StudID is the primary key, the key, No, E-mail becomes an alternate key. StudID Roll No First Name LastName Email 1 11 Tom Price This email address is being protected from spambots. You need JavaScript enabled for viewing. 2 12 Nick Wright This email address is being protected from spambots. You need JavaScript enabled for viewing. 3 13 Days Natan This email address is being protected from spambots. You need JavaScript enabled for viewing. CANDIDATE KEY is a set of attributes that uniquely identify the tuples in a table. Candidate Key is a super key without repeated attributes. The primary key should be selected from among the keys of the candidate. Each table must have at least one candidate key. A table can have multiple candidate keys, but only one primary key. Candidate Key Properties: Must contain unique ValuesKandidate key can have multiple attributesMust does not contain null valuesIt should contain minimal fields to ensure uniquenessConsidentically identify each record in the TableIsdample: In the default Table Stud ID, Roll No, and e-mail are the keys of the candidates who help us to uniquely identify the student record in the table. StudID Roll No First Name LastName Email 1 11 Tom Price This email address is being protected from spambots. You need JavaScript enabled for viewing. 2 12 Nick Wright This email address is being protected from spambots. You need JavaScript enabled for viewing. 3 13 Days Natan This email address is being protected from spambots. You need JavaScript enabled for viewing. A FOREIGN KEY is a column that creates a relationship between two tables. The purpose of foreign keys is to maintain data integrity and enable navigation between two different entity cases. It acts as a cross-reference between two tables because it refers to the primary key of the second table. Example: DeptCode DeptName 001 Science 002 English 005 Computer Teacher ID Fname Lname B002 David Warner B017 Sara Joseph B009 Mike Brunton In this key in the example of DBMS we have two desks, teaching and a department in school. However, there is no way to see which search job in which department. In this table, by adding a foreign key to Deptcode in the name of the Master, we can create a relationship between the two tables. DeptCode Fname Fname B002 002 teacher David Warner B017 002 Sara Joseph B009 001 Mike Brunton This concept is also known as reference integrity. COMPOUND KEY has two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique in itself within the database. However, in combination with another column or columns, the combination of composite keys becomes unique. The purpose of the merge key in the database is to uniquely identify each record in the table. Example: OrderNo PorductID Product Name Quantity B005 JAP102459 Mouse 5 B005 DKT321573 USB 10 B005 OMG446789 LCD Monitor 20 B004 DKT321573 USB 15 B002 OMG446789 Laser Printer 3 In this example, OrderNo and ProductID cannot be the primary key because they do not uniquely identify the record. However The Order ID and Product ID key can be used because it uniquely identifies each record. A COMPOSITE KEY is a combination of two or more columns that uniquely identify rows in a table. The combination of columns guarantees uniqueness, although individual uniqueness is not guaranteed. Therefore, they are combined to uniquely identify the records in the table. The difference between a join and a composite key is that any part of a connecting key can be a foreign key, but a composite key may or may not be part of a foreign key. SURROGATE KEYS is an artificial key that aims to uniquely identify each record called a surrogate key. This type of partial key in dbms is unique because it is created when you do not have any natural primary key. They do not give any meaning to the data in the table. A replacement key is usually a utier. A replacement key is a value generated immediately before a record is inserted into a table. Fname Lastname Start time Anne Smith 09:00 18:00 Jack Francis 08:00 17:00 Anna McLean 11:00 20:00 Shown Willam 14:00 23:00 Above, given for example, the displayed shift time of the different employee. In this example, a replacement key is required to uniquely identify each employee. Alternate keys in m2 are allowed when no property has a primary key parameter. In a table when the primary key is too large or complicated. The difference between the primary key and the foreign keyPrimary The key key helps you uniquely identify the record in the table. This is the field in the table that is the primary key of the second table. The primary key never accepts null and void values. A foreign key can accept multiple zero values. The primary key is a grouped index, and the data in the DBMS table is physically organized in the sequence of the grouped index. A foreign key cannot automatically create an index, grouped, or not grouped. However, you can manually create an index on a foreign key. You can have one primary key in the table. You can have multiple foreign keys in the table. Summary Key in SQL is an attribute or set of attributes that helps you identify the row (tuple) in the relationship(table) DBMS keys allow you to establish a relationship between and identify the relationship between tables Seven types of DBMS keys are Super, Primary, Candidate, Alternate, Side, Join, Composite, and Surrogacy Key. A super key is a group of individual or multiple keys that identifies rows in a table. Column or group of columns in a table that helps us uniquely identify each row in that table is called the primary key All keys that are not the primary key are called an alternate key A super key without a repeat attribute is called a candidate key The merge key is a key that has many fields that allow you to uniquely recognize a specific key record that has multiple attributes for unique row recognition in the table is called the Composite Key Artificial Key that aims to uniquely identify each the primary key is called a replacement key never accepts null and void values until key can accept more null and void values. Page 6 Functional Dependency (FD) determines the relationship of one attribute to another attribute in the Database Management System (DBMS). Functional dependency helps you maintain the quality of your data in your database. Functional dependency is indicated by the arrow →. Functional dependence X on Y represents X → Y. Functional dependency plays a vital role in finding the difference between good and poor database design. Example: Number of employees Employee name Salary City 1 Dana 50000 San Francisco 2 Francis 38000 London 3 Andrew 25000 Tokyo In this example, if we know the value of the number of employees, we can get the employee name, city, salary, etc. With this we can say that the city, employee name and salary functionally depend on the number of employees. In this guide you will learn: Key conceptsSealed some key concepts for functional dependency: Description of key terms Axiom Axioms is a set of reasoning rules used to deduce all functional dependencies on the relational database. Decomposition This is a rule that suggests that if you have a table that seems to contain two entities that are determined by the same primary key, then you should consider breaking them into two different tables. Dependent It appears on the right side of the functional dependency diagram. The option is displayed on the left side of the functional dependency diagram. The Union proposes that if two tables are separated and the PK is the same, you should consider putting them. Together The rules of functional dependencies There are three most important rules for functional dependency: the reflexive rule –. If X is a set of attributes and Y is_subset_of X, then X has a value of Y. Magnification rule: When x -> y holds, and c is an attribute set, then ac -> bc also holds. This is adding attributes that do not change basic dependencies. Transitivity Rule: This rule is very similar to the transient rule in algebra if x -> y holds and y -> z holds, then x -> z also holds. X -> y is called as functional that determines y. Types of functional dependenciesMultivalued dependency: Trivial functional dependency: Non-trivial functional dependency: Multiple dependency in DBMSMultivalued dependencies occurs in a situation where there are multiple independent multiple attributes in one table. Multiple dependency is a complete limitation between two sets of attributes in a relationship. It requires that certain tuples be present in the relationship. Example: Car_model Maf_year Color H001 2017 Metallic H001 2017 Green H005 2018 Metallic H005 2018 Blue H010 2015 Metallic H033 2012 Grey In this example, maf_year and color are independent of each other but dependent on car_model. In this example, these two columns are said to be multi-dependent on car_model. This addiction can be presented like this: car_model -> maf_year car_model-> trivial functional dependency:Trivial dependency is A set of attributes called trivial if an attribute set is included in that attribute. So, X -> Y is a trivial functional dependency if Y is subset X. For example: Emp_id Emp_name AS555 Harry AS811 George AS999 Kevin Consider this two-column table Emp_id and Emp_name. {Emp_id, Emp_name} -> Emp_id is a trivial functional dependency because Emp_id subset {Emp_id.Emp_name}. No trivial functional dependence on DBMSFunctional dependency which is also known as nontrivial dependence occurs when A->B is valid where B is not subset A. In a relationship, if attribute B is not a subset of attribute A, then it is considered a non-trivial dependence. Age Microsoft CEO Satya Nadella 51 Google Sundar Pichai 46 Apple Tim Cook 57 Example: (Company} -> {CEO} (if we know the Company, we know the CEO's name) But CEO is not a subset of the company, and therefore is a non-trivial functional dependence. Transient dependence:It is a transient type of functional dependence that occurs when t indirectly forms two functional dependencies. Example: Age Microsoft CEO Satya Nadella 51 Google Sundar Pichai 46 Alibaba Jack Ma 54 {Company} -> {CEO} (if we know compay, we know the name of its CEO) {CEO } -> {Age} If we know the CEO, we know age Therefore according to the rule rule of transient dependency: { Company} -> {Age} should be held, it makes sense because if we know the name of the company , we can know his age. Keep in mind: You need to remember that transient addiction can only occur in relation to three or more attributes. What is normalization? Normalization is a method of organizing data into a database that helps you avoid data redundancy, insertion, updating, and deleting anomalies. It is a process of analyzing the relationship schemes based on their different functional dependencies and primary key. Normalization is inherent in relational database theory. This can have the effect of duplicating the same data within the database, which can result in additional tables being created. Benefits of functional dependencyFunctional dependency avoids data redundancy. Therefore, the same data is not repeated in multiple locations in this databaseIt helps you maintain the quality of the data in the databaseIt helps you define the meanings and limitations of the databaseIt helps you identify bad designsIt helps you find the facts regarding the design of the databaseSummary Functional Dependency is when one attribute determines another attribute in the DBMS system. Axiom, decomposition, depending on, determinant, Union are key concepts for functional dependency Four types of functional dependencies are 1) Multiple 2) Trivial 3) Non-trivial 4) Transient multiple dependency occurs in a situation where there are multiple independent multiple attributes in one table Trivial dependency occurs when a set of attributes called trivial if a set of attributes is included in this attribute non-life dependency occurs when A-gt;B Truth where B is not subset A is a transitory type of functional dependency that occurs when indirectly formed with two functional dependencies Normalization is a method of organizing data into a database that helps you avoid data redundancy Page 7Data Independence is defined as the property of DBMS that helps you change the database schema at one level of the database system without having to change the schema at the next higher level. Data independence helps you keep your data separate from all programs that use it. You can use this stored data for computing and presentation. In many systems, data independence is an essential function for system components. In this guide you will find out: Data independence types In DBMS there are two types of data independence Independence of physical dataLogic data independence. DatabaseBee levels teach data independence, refreshing database levels is important. The database has 3 levels as shown in the diagram below the physical/InternalConceptualExternal level of the DBMS architecture diagram to consider an example of a university database. At different levels, this is what implementation will look like: Schema Deployment Type External Schema View 1: Info Course(cid:int,cname:string) View 2: novnfo(id:int. name:string) Conceptual Scheme Students(id: int, name: string, login: string, age: integer) Courses(id: int, cname.string, credits:integer) Enrolled(id: int, grade:string) Physical relationship schemes stored as unocued files. The index on the first students.Physical Data IndependencePhysical data independence column helps you separate conceptual levels from the internal/physical level. Allows you to provide a logical description of the database without having to specify physical structures. Compared to logical independence, it is easy to achieve the independence of physical data. With physical independence, you can easily change physical storage structures or devices with an effect on the conceptual scheme. Any change made would be absorbed by mapping between conceptual and internal levels. The independence of physical data is achieved by the presence of the internal level of the database, and then by transforming it from the conceptual level of the database to the internal level. Examples of changes in Physical Data IndependenceDue to Physical independence, any of the changes below will not affect the conceptual layer. Using a new storage device such as Hard Disk or Magnetic StripsModifying technique of organizing files in the DatabaseSwitching on different data structures. Change the way you access it. I'm changing indices. Changes to compression techniques or hash algorithms. Changing the location of the database from, say, C drive to D DriveLogical Data IndependenceLogical Data Independence is the ability to change the conceptual schema without changingexternal viewsExternal API or programsOur change will be absorbed by mapping between external conceptual levels. Compared to the independence of physical data, it is challenging to achieve logical data independence. Examples of changes in the Logical Data IndependenceDue logical independence box, any of the changes below will not affect the outer layer. Add/Modify/Delete a new attribute, entity, or relationship is possible without rewriting existing ApplicationMerging two records into a singleBreaking existing record in two or more recordsDifference between physical and logical data security Data independence Data independence Data independence Logical data Independence mainly deals with the structure or change of data definition. It mainly refers to data storage. It is difficult because retrieving data depends mainly on the logical structure of the data. It's easy to reach. Compared to the logic Physical independence is difficult to achieve logical data independence. Compared to logical independence, it is easy to achieve the independence of physical data. You must make changes to the Application program if new fields are added or deleted from the database. Changing the physical level usually does not need to be changed at the Application level. Modification at logical levels is significant whenever the logical structures of the database change. Changes at internal levels may or may not be necessary to improve the performance of the structure. Deals with a conceptual scheme that deals with the internal scheme by example: Add / modify / Delete / Delete a new example: change in compression techniques, algorithms for hash, Storage devices, etc. The importance of data independenceSears to improve the quality of dataDatabase maintenance becomes affordableInstising standards and improving database security You do not need to change the structure of data in application programsPermit developers to focus on the general structure of the Database instead of worrying about internal implementation It allows you to improve the state that is undivided or undividedDatabase is inconsistently reduced. Improving system performance requires simple changes at the physical level. SummaryData Independence is the property of DBMS, which helps you change the database schema at one level of the database system without having to change the schema at the next higher level. Two levels of data independence are 1) Physical and 2) Logical physical independence of data helps you separate conceptual levels from internal/ physical levelsLogic data independence is the ability to change the conceptual scheme without changingSas compared to the independence of physical data, it is challenging to achieve logical data independenceData Independence helps you to improve the quality of dataPage 8In DBMS, hashing is a technique for directly searching the location of desired data on the disk without using the structure of the index. The hashish method is used to index and retrieve items in the database Search this particular item with a shorter key instead of using its original value. Data is stored in the form of data blocks whose address is generated by applying the hash function at the location of memory where these records are stored known as a data block or data bucket. In this DBMS guide, you will learn, why do we need Hashing? Here are the situations in DBMS where you need to apply the Hashing method:For a huge database structure, it is difficult to search all the index values throughout its level and then you need to reach the destination data block to get the data you want. The hashish method is used to index and retrieve items in a database because it is faster to search for that particular item using a shorter key instead of using the original value. Hashing is the ideal method for calculating the direct location of data records on a disk without using the index structure. It is also a useful technique for the implementation of dictionaries. Important terminologies used in HashingHere are important terminologies used in Hashing: Data bucket – Data buckets are memory locations where records are stored. It is also known as the Storage Unit. Key: A DBMS key is an attribute or set of attributes that helps you identify a row(tuple) in a relationship (table). This allows you to find a relationship between the two tables. Hasha function: The hash function is a mapping function that maps all expensive search keys to the address where the actual records are set. Linear sounding – Linear sounding is a fixed interval between probes. In this method, the next available data block is used to enter a new record, instead of overisclating it on an older track. Quadratic Sounding- Helps you determine the new address of the bucket. Helps you add the interval between probes by adding a consecutive square polynomic output to the initial value provided by the original calculation. Hash Index – This is the address of the data block. Hashish function can be a simple mathematical function even to a complex mathematical function. Double Hashing – Double hashing is a computer programming method used in hash tables to solve problems having a collision. Bucket dressing: The overflow state of the bucket is called a collision. This is a fatal phase for any static that has to work. There are mostly two types of SQL hashing methods: Static hashing Dynamic HashingStatic HashingIn static hashing, the resulting data bucket address will always remain the same. Therefore, if you generate an address for say Student_ID = 10 using the hash mode(3), the resulting address of the bucket will always be 1. So you will not see any change in the address of the bucket. Therefore, in this static hashish method, the number of data buckets in memory always remains constant. Static hash functionsInserting record: When a new record requires to be inserted into a table, you can generate an address for a new record using the hash key. When an address is generated, the record is stored at that site. Search: When you need to retrieve a record, the same hashish function should be useful for retrieving the address of the bucket where the data should be stored. Delete a record: You can use the hashish function to retrieve the record that you want to delete. You can then remove the records for that address in memory. Static hashish is further divided into Open hashing Close hashing. Open HashingIn Open hashing method, Instead of rewriting the older next available data block used to enter a new record, this method is also known as linear sounding. For example, A2 is the new record that you want to insert. The hashish function generates the address as 222. But it's already occupied with some other value. That's why the system is looking for the next 501 data bucket and assigning it A2. As Open Hash works Close HashingIn a close method of hashish, when the buckets are full, a new bucket is allocated for the same hashish, and the result is linked after the previous one. Dynamic HashingDynamic hashing offers a mechanism in which data buckets are added and removed dynamically and on demand. In this hashing, the hashish function helps you create a large number of values. Compare ordered indexing and hashingParameters order indexing hashing storage address addresses in memory sort by a key value called primary key Addresses are always generated using the hash function to a key value. Performance Can be reduced when data is increased in a hash file. Because it stores data in a sorted format when any operation is performed (insert/delete/update), which reduces its performance. Hashish performance will be best when there is constant addition and deletion of data. However, when the database is huge, then the organization of hash files and its maintenance will be more expensive. Use Preferred to retrieve data in a range– meaning that whenever there is fetch data for a specific range, this method is the ideal option. This is the ideal method when you want to retrieve a specific record based on a search key. However, this will only be good when the hash function is on the search key. Memory Management There will be many unused data blocks due to the delete/update operation. These data blocks cannot be published for reuse. That is why regular memory maintenance is required. In static and dynamic hash methods, memory is always managed. The bucket dressing is also perfectly handled to spread static hashish. What is Sudar? A hashish collision is a condition when the resulting hashish from two or more data in a data set incorrectly maps the same location in the hash table. How to deal with Hashing Collision? There are two techniques that you can use to avoid a hash collision: Rehashing: This method, invokes the secondary function of hashish, which is applied continuously until an empty groove is found, where the record should be placed. Chain chain: The chain method builds a related list of items whose key hashishes are the same This method requires an additional field to link to each table position. Summary: In DBMS, hashing is a technique for directly searching the location of the desired data on the disk without using the index structure. The hashish method is used to index and retrieve items in a database because it is faster to search for that particular item using a shorter key instead of using the original value. Data bucket, Key , Hash function, Linear Sounding, Quadratic probing, Hash index, Double Hashing, Bucket Overflow are important terminologies used in hashing Two types of hashish methods are 1) static hashing 2) dynamic hashing In static hashish, the resulting data bucket address will always remain the same. Dynamic hashing offers a mechanism in which data buckets are added and removed dynamically and on demand. In order indexing addresses in memory are sorted by critical value, while in hash addresses they are always generated using the hash function at a key value. A hashish collision is a condition when the resulting hashish from two or more data in a data set incorrectly maps the same location in the hash table. Rehashing and chaining are two methods that help you avoid a hash collision. Page 9SQL is a database language designed to retrieve and manage data into a relational database. SQL is the standard language for managing a database. All RDBMS systems such as MySQL, MS Access, Oracle, Sybase, Postgres, and SQL Server use SQL as their standard database language. In this DBMS tutorial, you will learn: Why use SQL? Here are important reasons to use SQL Helps users access data in the RDBMS system. It helps you describe the data. Allows you to define data in a database and manipulate that specific data. With SQL, you can create and drop databases and tables. SQL offers you to use the function in the database, create a view, and process stored. You can set permissions on tables, procedures, and views. A brief history of SQLHera, are important landmarks from the history of SQL: 1970 - Dr. Edgar F. Ted Codd described the relational model for databases. 1974 . 1978 - IBM released a product called System/R. 1986 - IBM developed a prototype relational database, standardized by ANSI. 1989 - First version launched since SQL 1999 - SQL 3 launched with features such as triggers, object orientation, etc. SQL2003 window functions, XML-related features, etc. SQL2006- SUPPORT for XML Query Language SQL2011 Enhanced Time Database Support SQLHere are five types of widely used SQL queries. Data Definition Language (DDL)Data Manipulation Language (DML)Data Control Language (DCL)Transaction Control Language (TCL)Data Query Language (DQL) Let him see each of them in detail: What is DDL? The language of the data definition helps you define the database structure or schema. The five types of DDL commands are: CREATECREATE statements are used to define the base structure schema Sintaksa: CREATE TABLE TABLE (COLUMN_NAME DATATYPES[,....]); For example: Create a university database; Creating a table of students; Create a for_students; DROPDrops commands remove tables and databases from RDBMS. Syntax Drop Table ; For example: Drop object_type object_name; Drop database university; Student with drop-down table; The ALTERAlters command allows you to change the structure of the database. Syntax: To add a new column to the ALTER TABLE table_name add column_name column definition; To modify an existing column in a table: ALTER TABLE MODIFY; For example: Alter table guru99 add subject varchar; TRUNCATE: This command is used to delete all rows from the table and free up space containing the table. Syntax: TRUNCATE STOL table_name; Example: TRUNCATE table students;Data Manipulation Language (DML) allows you to modify a database instance by inserting, modifying, and deleting its data. He is responsible for making all kinds of data changes to the database. There are three basic constructs that allow the database program and the user to enter data, and the information is: Here are some important DML statements: INSERT: This statement is an SQL query. This command is used to insert data into a table row. Syntax: INSERT into TABLE_NAME (col1, col2, col3,.... col N) VALUES (value1, value2, value3, ... valueN); Or INSERT into TABLE_NAME value (value1, value2, value3, ... valueN); For example: INSERT INTO students (RollNo, FIrstName, LastName) VALUES ('60', 'Tom', Erichsen'); UPDATE: This command is used to update or modify column values in a table. Syntax: TABLE_NAME SET [column_name1= value1,... column_nameN = valueN] [WHERE IS THE STATUS] For example: UPDATE students SET FirstName = 'Jhon', LastName= 'Wick' WHERE Studid = 3; DELETE: This command is used to remove one or more rows from the table. Syntax: DELETE TABLE_NAME [WHERE is the status]; For example: DELETE FROM STUDENTS WHERE FirstName = 'Jhon'; DCL includes commands such as GRANT and REVOKE, which are useful for granting rights and permissions. Other permissions control the parameters of the database system. Examples of DCL commands:Commands that come under DCL: Grant: This command is used to grant user access privileges to the database. Syntax: GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER; For example: GRANT SELECT ON Users TO'Tom'@'localhost; Revocation: It is useful to restore user permissions. Syntax: RECALL privilege_nameON object_nameFROM {user_name | PUBLIC |role_name}For example: REVOCATION OF SELECTION, UPDATE ON STUDENT FROM BCA, MCA; The transaction control language or TCL commands deal with a transaction within the database. The CommitThis command is used to store all transactions in the database. Syntax: Oblige; For example: DELETE FROM STUDENTS WHERE RollNo =25; COMMIT; RollbackRollback command allows transactions that have not already been saved to the database. Syntax: ROLLBACK; Example: DELETE FROM STUDENTS WHERE RollNo =25; =25; The command helps you set the save point within the transaction. Syntax: SAVEPOINT SAVEPOINT_NAME; Example: SAVEPOINT RollNo; The data query language (DQL) is used to retrieve data from a database. Use only one command: SELECT: This command helps you select an attribute based on the condition described in the WHERE clause. Syntax: SELECT EXPRESSIONS FROM TABLES WHERE CONDITIONS; For example: CHOOSE The first name name from a student where rollno > 15; Summary:SQL is the language of the database intended for retrieving and managing data into a relational database. It helps users access data in the RDBMS system In the year 1974. Term Structured Query Language appeared Five types of SQL queries are 1) Data Definition Language (DDL) 2) Data Manipulation Language (DML) 3) Data Control Language (DCL) 4) Transaction Control Language (TCL) and, 5) Data Collection Language (DQL) Data Definition Language (DDL) helps you define database structure or schema. Data Manipulation Language (DML) allows you to modify a database instance by inserting, modifying, and deleting its data. DCL includes commands such as GRANT and REVOKE, which are useful for granting rights and permissions. The transaction control language or TCL commands deal with a transaction within the database. The data query language (DQL) is used to retrieve data from a database. Page 10Join in DBMS is a binary operation that allows you to combine product combination and selection in a single statement. The goal of creating association conditions is to help you combine data from two or more DBMS tables. Tables in DBMS are linked using the primary key and foreign keys. In this DBMS guide, you will learn: In DBMS, there are mostly two types of associations: Theta, Natural, EQUI Outer Join: Left, Right, FullLet's see them in detail: INNER JOIN is used to restore rows from both tables that meet the default state. It is the most commonly used association operation and can be considered the default associated type An Inner join or equijoin is a comparative assembly that uses equality comparisons in join-predicate. However, if you use other comparison operators such as > this cannot be called equijoin. Inner Join further divided into three sub-subtites: Theta joinNatural joinEQUI joinTHETA JOIN allows you to merge two tables based on the state represented by theta. Theta joins the work for all comparison operators. It is marked with a θ. The general case of OPERATION JOIN is called Theta join. Syntax: ⋈θ BTheta can use all conditions in the selection criteria. Consider the following tables. Table B Column 1 Column 2 Column 1 Column 2 1 1 1 1 1 1 2 1 3 For example: ⋈ column 2 > B.column 2 (B)Column ⋈ column 2 > column 2 (B) column 1 column 2 1 2 EQUI JOIN is done when the Theta port uses only the equivalence condition. Joining EQUI is the most difficult operation to carry out effectively in RDBMS and one of the reasons why RDBMS has significant problems with For example: example: ⋈ column 2 = B.column 2 (B)A ⋈ A.column 2 = B.column 2 (B) column 1 column 2 1 1 NATURAL JOIN is not used by any of the comparison operators. In this type of association, attributes should have the same name and domain. In Natural Join, there should be at least one common attribute between the two relationships. It performs the choice of forming equalitys on those attributes that appear in both relationships and eliminates duplicate attributes. Example: Consider the following two tables C ⋈ DC ⋈ D Num Square Cube 2 4 8 3 9 18 OUTER JOIN does not require each record in the two merge tables to have the appropriate record. In this join type, the table retains each record even if there is no other corresponding record. The three types of outer joins are: Left Outer JoinRight External JoinFull Outer JoinLEFT JOIN returns all rows from the table on the left, even if no corresponding rows were found in the table on the right. When the corresponding record is not found in the table to the right, null is returned. Consider the following 2 tables A Num Square 2 4 3 9 4 16 A BA ⋈ B Num Square Cube 2 4 8 3 9 18 4 16 - RIGHT JOIN returns all columns from the table on the right, even if no corresponding rows were found in the table on the left. Where no matches were found in the table on the left, NULL returns. RIGHT External JOIN is the opposite of LEFT JOIN In our example let's assume that you need to get the members' names and movies that they rent. Now we have a new member who hasn't rented any of the film yet. A BA ⋈ B Num Cube Square 2 8 4 3 18 9 5 75 - IN FULL OUTER JUNCTION , all tuples from both relationships are included in the result, regardless of the appropriate condition. Example: BA ⋈ B Num Square Cube 2 4 8 3 9 18 4 16 - 5 - 75 Summary: In DBMS 1) Inner Join 2 there are generally two types of joining. Inner Join is further divided into three subtites: 1) Theta join 2) Natural join 3) EQUI join Theta Join allows you to connect two tables based on the state represented by theta When theta join uses only the equi join condition. The natural port is not used by any of the comparison operators. External Association does not require each record in the two merge tables to have the appropriate record. Outer Join is further divided into three subtites: 1)Left outside join 2) Right outside join 3) Full Outer Join The LEFT Outer Join returns all rows from the table on the left, even if no corresponding rows were found in the table on the right. RIGHT Outer Join returns all columns from the table to the right, even if no corresponding rows were found in the table on the left. In a complete outer compound, all tuples from both relationships are included in the result, regardless of the appropriate condition. Page 11INDEXING is a data structure technique that allows you to quickly retrieve records from a database file. An index is a small table that has only two columns. First consists of a copy of the primary or candidate key table. Its second column contains a set of pointers to hold the address of the disk block where this specific key value is stored. Index - Takes the search key because the inputEffix returns a collection of corresponding records. In this DBMS indexing guide, you will learn: The indexing types of the database index type are defined based on its indexing attributes. The two main types of indexing methods are: Primary IndexingSecondary IndexingPrimary Indexing Indexing Indexing Indexing Index is a commissioned file that is a fixed length size with two fields. The first field is the same primary key, and the second, submitted is pointed to that particular data block. In primary Index, there is always one to one relationship between entries in an index table. Primary indexing is also further divided into two types. A dense IndexU dense index creates a record for each search key valued in the database. This helps you search faster, but you need more space to store index records. In this index, the method records contain the value of the lookup key and indicate the actual record on the disk. Rare IndexIt is an index record that appears only for some of the values in the file. A rare index helps you solve problems of dense indexing. In this method of indexing technique, a series of index columns stores the same data block address, and when data needs to be retrieved, the block address will be reached. However, a rare index stores index records for only some search key values. It needs less space, less overhead maintenance costs to insert and delete, but it's slower compared to the dense Index for locating records. An example of a secondary index of a rare index Secondary Index can generate a field that has a unique value for each record and should be the candidate key. It is also known as the non-grouping index. This two-level database indexing technique is used to reduce level 1 mapping size. For the first level, a large range of numbers is selected as a result; the size of the mapping always remains small. Example of secondary indexing In a bank account database, data is stored sequentially acc_no; you may want to find all the accounts in a particular ABC Bank branch. Here you can have a secondary index for each search key. An index record is a record point on a bucket that contains pointers to all

records with their specific search key value. Grouping indexU grouped index, the records themselves are stored in the Index, not the cursor. Sometimes an index is created on non-primary columns, which may not be unique to each record. In such a situation, you can group two or more columns to get unique values and create an index called a grouped Index. This also helps you identify the record faster. Example: Suppose that company employed many employees in different departments. In this case, cluster indexing should be created for all employees belonging to the same It is considered in one cluster, and index points indicate a cluster as a whole. Here, the _no is some unique key. What is the Multilevel Index? Multilevel indexing occurs when the primary index does not fit in memory. In this type of indexing method, you can reduce the number of disk accesses to shorten any record and keep it on disk as a sequential file and create a rare base on that file. The B-Tree IndexB-tree index is a widely used indexing data structure. It is a multileash index format technique that is a balanced binary search tree. All Adversal Nods B of the tree indicate actual data indicators. Moreover, all adhesive nods are interconnected with the list of links, which allows the B tree to support a random and sequential approach. Lead nodes must have values between 2 and 4. Each time from the root to the leaf is mostly at the same length. Non-cissive nods other than the root nod have between 3 and 5 children's nods. Any nod other than root or leaf has between n/2] and n children. Advantages of indexing Important advantages / advantages of indexing are: It helps you reduce the total number of I/O operations needed to retrieve this data, so you do not need to access the queue in the database from the index structure. It offers users faster search and data retrieval. Indexing also helps you reduce table space because you don't need to bind to a row in a table because there's no need to store ROWID in the Index. This will help you reduce the space at the table. You cannot sort data into lead vents because the primary key value classifies it. The disadvantages of IndexingImportant deficiencies/counter Indexing are: To perform a database indexing management system, you need a primary key on a table with a unique value. You cannot perform other indexes on index data. You are not allowed to split a table organized by an index. SQL Indexing Decrease performance in INSERT, DELETE, and UPDATE query. Summary: Indexing is a small table consisting of two columns. The two main types of indexing methods are 1)Primary indexing 2) Secondary indexing. The primary index is an ordered file that is a fixed-sized file with a length of two fields. The primary index is also further divided into two types 1)Dense index 2)Sparse Index. A record is created in a dense index for each search key valued in the database. A rare method of indexing helps you solve problems of dense indexing. A secondary index is an indexing method whose search key determines the order in which it differs from the sequential order of the file. The grouping index is defined as an order data file. Multilevel indexing occurs when the primary index does not fit in memory. The biggest advantage of Indexing is that it helps you reduce the total number of I/O operations needed to retrieve this data. The biggest drawback in the performance of database management systems you need a primary key on a table with a unique value. Page 12 Details Last updated: 07 October 2020 DBMS is software used for storage and and Data. DBMS was introduced during the 1960s to store any data. It also offers data manipulation such as inserting, deleting, and updating data. The DBMS system also performs functions such as defining, creating, auditing, and controlling databases. It is specifically designed to create and maintain data and enable the individual business application to extract the desired data. What is RDBMS? The Relational Database Management System (RDBMS) is an advanced version of the DBMS system. It was created during the 1970s. The RDBMS system also allows the organization to access data more efficiently than DBMS. RDBMS is a software system that is used to store only data that needs to be stored as tables. In this type of system, data is managed and stored in rows and columns that are known as tuples and attributes. RDBMS is a powerful data management system and is widely used worldwide. DBMS stores data as a file, while in RDBMS, data is stored as tables. DBMS supports individual users, while RDBMS supports multiple users. DBMS does not support client and server architecture, but RDBMS supports client and server architecture. DBMS has low software and hardware requirements, while RDBMS has higher hardware and software requirements. In DBMS, data redundancy is common while in RDBMS, keys and indices do not allow data redundancy. The difference between DBMS vs RDBMSParameter DBMS RDBMS Storage DBMS stores data as a file. Data is stored as tables. The structure of the DBMS system database stores data in navigational or hierarchical form. RDBMS uses a tabulas structure with column names in the headers and rows containing the corresponding Number of DBMS users values supporting only one user. Supports multiple users. ACID In a regular database, data cannot be stored by acid model. This may develop inconsistencies in the database. Relational databases are harder to build, but they are consistent and well structured. They listen to ACID (Atomicity, Consistency, Isolation, Durability). Type of program It is a program for managing databases on computer networks and hard drives of the system. These are database systems used to maintain table relationships. Needs for hardware and software. Low needs for software and hardware. Greater need for hardware and software. DBMS integrity restrictions do not support integrity constants. Integrity constants are not imposed at the file level. RDBMS supports schema-level integrity restrictions. Values outside the defined range cannot be stored in a specific RDMS column. Normalization of DBMS does not support normalization RDBMS can be normalized. Distributed databases DBMS do not support distributed databases. RBMS offers support for distributed databases. Ideally adapted for the DBMS system mainly deals with a small amount of data. RDMS is designed to handle large quantities Dr. E.F. Cod rules Dbms meet less than seven Ph.D. Sc. Cod rules Dbms meet 8 to 10 Dr. E.F. Codd Client Server DBMS policy does not support client architecture and server RDBMS supports client and server architecture. Retrieving data is slower for complex and large amounts of data. Retrieving data is quick because of its relational approach. Redundancy of data Redundancy of data is common in this model. Keys and indexes do not allow data redundancy. Data Relationship No data relationship Data is stored in the form of tables that are interconnected by foreign keys. Insurance No insurance. Multiple levels of security. Log files are created at the OS, Command, and Object level. Data access data elements must be accessed individually. Data can be easily accessed using an SQL query. Multiple data elements can be accessed at the same time. Examples Examples of DBMS are file system, XML, Windows registry, etc. An example of RDBMS is MySQL, Oracle, SQL Server, etc. Group by Clause is an SQL statement used for a group of rows that... Read more SQLite databases are very lightweight. Unlike other database systems, there is no configuration,... Read more What are the decision-making statements? Decision-making statements are the ones that will decide... Read more What is MySQL? MySQL is an open source relational database. MySQL is a cross platform which means that ... Read more In this guide, you will learn- SQLite limit Primary key Is not null and null limit DEFAULT ... Read more Now that Myflixdb is, what's next? Congratulations for your success completing SQL tutorial ... Read more

aha acls guideline pdf , 36856427634.pdf , letter_to_daughter_going_to_college_from_dad.pdf , careercup cracking the coding interview pdf , zijaritun.pdf , interesting brain teasers with answe , smacna sheet metal manual pdf , analyse_entretien_semi_directif.pdf , homm 3 hd mod gog , assumptions of ols pdf , aparejos en construccion pdf , cabbage family vegetables , temperament test with interpretation pdf , intersecting_and_parallel_lines_worksheet.pdf ,