**RDBMS CONCEPTS**

**What is database?**A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.  
  
**What is DBMS?**

* It is a collection of programs that enables user to create and maintain a database.
* In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

**What is the purpose of DBMS?**

* It supports for managing users and processing SQL queries, performing backups/restores and associated tasks.
* RDBMS usually include an API so that developers can write programs that use them.
* Typical RDBMS include Microsoft Access, Microsoft SQL Server, Sybase, IBM DB2, Oracle, Ingres, My SQL, and many more.

**What are E.F. Codd’s Rules?**

These rules are proposed by Edgar F. Codd, a pioneer of the relational model for databases, designed to define what is required from a database management system in order for it to be considered relational, i.e., a relational database management system (RDBMS).

**Codd Rules:**

* **Rule 0:** The system must qualify as relational, as a database, and as a management system.
* **Rule 1:** The information rule: All information in the database is to be represented in one and only one way, namely by values in column positions within rows of tables.
* **Rule 2:** All data must be accessible. This rule is essentially a restatement of the fundamental requirement for primary keys. It says that every individual scalar value in the database must be logically addressable by specifying the name of the containing table, the name of the containing column and the primary key value of the containing row.
* **Rule 3:** The DBMS must allow each field to remain null (or empty). Specifically, it must support a representation of "missing information and inapplicable information" that is systematic, distinct from all regular values (for example, "distinct from zero or any other number", in the case of numeric values), and independent of data type. It is also implied that such representations must be manipulated by the DBMS in a systematic way.
* **Rule 4:** The system must support an online, inline, relational catalog that is accessible to authorized users by means of their regular query language. That is, users must be able to access the database's structure (catalog) using the same query language that they use to access the database's data.
* **Rule 5:**

The system must support at least one relational language that   
1. Has a linear syntax  
2. Can be used both interactively and within application programs.  
3. Supports data definition operations (including view definitions), data manipulation operations (update as well as retrieval), security and integrity constraints, and transaction management operations (begin, commit and rollback).

* **Rule 6:** All views that are theoretically updatable must be updatable by the system.
* **Rule 7:** The system must support set-at-a-time insert, update, and delete operators. This means that data can be retrieved from a relational database in sets constructed of data from multiple rows and/or multiple tables. This rule states that insert, update, and delete operations should be supported for any retrievable set rather than just for a single row in a single table.
* **Rule 8:** Changes to the physical level (how the data is stored, whether in arrays or linked lists etc.) must not require a change to an application based on the structure.
* **Rule 9:** Changes to the logical level (tables, columns, rows, and so on) must not require a change to an application based on the structure. Logical data independence is more difficult to achieve than physical data independence.
* **Rule 10:** Integrity constraints must be specified separately from application programs and stored in the catalog. It must be possible to change such constraints as and when appropriate without unnecessarily affecting existing applications.
* **Rule 11:** The distribution of portions of the database to various locations should be invisible to users of the database. Existing applications should continue to operate successfully:   
  1. when a distributed version of the DBMS is first introduced; and  
  2. when existing distributed data are redistributed around the system.
* **Rule 12:** If the system provides a low-level (record-at-a-time) interface, then that interface cannot be used to subvert the system, for example, bypassing a relational security or integrity constraint.

**What are the disadvantages of File Processing System?**

* No centralized control.
* Data Redundancy.
* Data Inconsistency.
* Data cannot be shared.
* Standards cannot be enforced.
* Security issues.
* Integrity cannot be maintained.
* Data dependence.

**What is RDBMS?**

A relational database management system (RDBMS) is a database management system (DBMS) that utilizes a relational database model to hold the data used in the application.

**What are the advantages of RDBMS?**

* Centralized control.
* No Data Redundancy
* Data Consistency
* Data can be shared
* Standards can be enforced
* Security can be enforced
* Integrity can be maintained
* Data independence

**Explain Table, Record, and Column?**

* **Table:** The basic construct of a relational database that contains rows and columns of related data.
* **Row:** A group of one or more data elements in a database table that describes a person, place, or thing.
* **Column:** The component of a database table that contains all of the data of the same name and type across all rows.

**Explain Entity, Entity Types and Entity Set?**

* **Entity:** It is a 'thing' in the real world with an independent existence.
* **Entity type:** It is a collection (set) of entities that have same attributes.
* **Entity set:** It is a collection of all entities of particular entity type in the database.

**What is relation?**

It is the number of attribute of its relation schema.

**Explain Relationship, Relationship Set, and Relationship Type?**

* **Relationship:** It is an association among two or more entities.
* **Relationship Set:** The collection (or set) of similar relationships.
* **Relationship Type:** Relationship type defines a set of associations or a relationship set among a given set of entity types.

**What are Integrity Rules?**

The following categories of the data integrity exist with each RDBMS:

* **Entity Integrity:** There are no duplicate rows in a table.
* **Domain Integrity:** Enforces valid entries for a given column by restricting the type, the format, or the range of values.
* **Referential integrity:** Rows cannot be deleted, which are used by other records.
* **User-Defined Integrity:** Enforces some specific business rules that do not fall into entity, domain, or referential integrity.

**What is constraint?**

* Constraints are the rules enforced on data columns on table.
* These are used to limit the type of data that can go into a table.
* This ensures the accuracy and reliability of the data in the database.
* Constraints could be column level or table level. Column level constraints are applied only to one column where as table level constraints are applied to the whole table.

**What is Normalization?**

It is a process of analyzing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties.

**What is the purpose of Normalization?**

Minimizes redundancy.

Minimizes insertion, deletion and update anomalies.

**What are different types of normal forms?**

* **First Normal Form (1NF):** The domain of attribute must include only atomic (simple, indivisible) values.
* **Second Normal Form (2NF):** A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.
* **Third Normal Form (3NF):** A relation schema R is in 3NF if it is in 2NF and for every FD X --> A either of the following is true  
  > X is a Super-key of R. > A is a prime attribute of R.

In other words, if every non prime attribute is non-transitively dependent on primary key.

* **Boyce-Code Normal Form (BCNF):** A relation schema R is in BCNF if it is in 3NF and satisfies additional constraints that for every FD X -> A, X must be a candidate key.
* **Fourth Normal Form (4NF):** A relation schema R is said to be in 4NF if for every multi valued dependency X --> Y that holds over R, one of following is true.  
  -X is subset or equal to (or) XY = R.  
  -X is a super key.
* **Fifth Normal Form (5NF):** A Relation schema R is said to be 5NF if for every join dependency {R1, R2... Rn} that holds R, one the following is true.

Ri = R for some I.

The join dependency is implied by the set of FD, over R in which the left side is key of R.

**What is De-Normalization?**

* De-normalization is the process of attempting to optimize the performance of a database by adding redundant data.
* It is sometimes necessary because current DBMSs implement the relational model poorly.
* A true relational DBMS would allow for a fully normalized database at the logical level, while providing physical storage of data that is tuned for high performance.
* De-normalization is a technique to move from higher to lower normal forms of database modeling in order to speed up database access.

**What is Data Model?**

A collection of conceptual tools for describing data, data relationships data semantics and constraints.

**What is ER-Modeling?**

* E-R model stands for Entity-Relationship model.
* This data model is based on real world that consists of basic objects called entities and of relationship among these objects.
* Entities are described in a database by a set of attributes.

**What is a Primary Key?**

Primary key is a key which uniquely identifies each row /records in a database table.

**What is a Foreign Key?**

Foreign key is a key which uniquely identified a rows/records in any another database table.

**What is Compound Key?**

* A compound key is a key that consists of 2 or more attributes that uniquely identify an entity occurrence.
* Each attribute that makes up the compound key is a simple key in its own right.

**What is Referential Integrity?**

* Referential integrity is to hold, any field in a table that is declared a foreign key can contain only values from a parent table's primary key or a candidate key.
* For instance, deleting a record that contains a value referred to by a foreign key in another table would break referential integrity. Some relational database management systems (RDBMS) can enforce referential integrity, normally either by deleting the foreign key rows as well to maintain integrity, or by returning an error and not performing the delete. Which method is used may be determined by a referential integrity constraint defined in a data dictionary.

**What is Index?**

* A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes and increased storage space.
* Indexes can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records.
* The disk space required to store the index is typically less than that required by the table (since indices usually contain only the key-fields according to which the table is to be arranged, and exclude all the other details in the table), yielding the possibility to store indices in memory for a table whose data is too large to store in memory.

Questions

What are the disadvantages of File Processing System?

What is Database and DBMS?

What are Disadvantages of DBMS?

What is RDBMS?

What are the advantages of RDBMS?

What are Integrity Rules?

What is constraint? Types of Constraints?

What is Normalization? Purpose of Normalization?

Types of Normal forms? What is De-normalization?

What is Referential Integrity?