**What are advantages of DBMS over traditional file-based systems?**

Database management systems were developed to handle the following difﬁculties of ﬁle-based systems supported by conventional operating systems.

1. Atomicity of updates. For each transaction, either all the operations are executed, or none of them.
2. Data redundancy and inconsistency.
3. Data isolation problems. If many transactions are running concurrently, it appears to the user that they are in a serial execution.
4. Integrity problems. Data should satisfy integrity constraints after the execution of transactions.
5. Accessing data concurrently.
6. Security problems.

**What are super, primary, candidate and foreign keys?**

**S**[**uperkey**](http://en.wikipedia.org/wiki/Superkey) is a set of attributes of a relation upon which all attributes of the schema are functionally dependent.

[**Candidate key**](http://en.wikipedia.org/wiki/Candidate_key) is a minimum superkey that no subset can form another key.

[**Primary Key**](http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/admin/c0004799.htm)is one of the candidate keys. There cannot be more than one primary keys in a table.

[**Foreign key**](http://en.wikipedia.org/wiki/Foreign_key) is a referential integrity constraint. It is a set of attributes in one table that uniquely identifies a tuple of another table. In other words, the values comes from a key of another table.

**What is the difference between primary key and unique constraints?**

Primary key cannot have NULL value, the unique constraints can have NULL values. There is only one primary key in a table, but there can be multiple unique constrains.

**What is database normalization?**

It is a process of analyzing the given relation schema based on their functional dependencies to achieve the following desired properties:  
1) Minimizing the data redundancy  
2) Minimizing the Insertion, Deletion, and Update anomalies  
Relation schemas that do not meet the properties are decomposed into smaller relation schemas that could meet the properties.

**Requirement for normalization**

1. No operation anomalies.
2. No information loss.
3. No constraint loss.

**What is BCNF and what problems does it have?**

A relation is in BCNF if for every nontrivial FD X->Y, X is the key.

A database is in BCNF if every relation in it is in BCNF.

BCNF guarantees information lossless but cannot guarantee dependency preserving (may lose some FDs).

**What is 3NF and what problems does it have?**

A relation is in 3NF if for every nontrivial FD X->Y, X is the key, or Y consists of prime attributs.

BCNF guarantees information lossless and dependency preserving, but may have operation anomalies.

**What is a good/bad database design?**

The relation schema has the following properties.

1. No operation anomalies.
2. No data redundancy.
3. No data loss.

If there is a FD applied on a table such that the set of attributes at the left side is not a key, then it is a bad schema. In other words, every FD should define a key of the table.

**What is SQL?**

SQL is Structured Query Language designed for defining, accessing and updating the [relational database system](http://en.wikipedia.org/wiki/Relational_database_management_system).

**What are the differences between DDL, DML and DCL in SQL?**

***DDL***stands for Data Definition Language. CREATE, ALTER, DROP and RENAME.  
***DML***stands for Data Manipulation Language. SELECT, INSERT and UPDATE.  
***DCL*** stands for Data Control Language. SQL queries like GRANT and REVOKE come under this.

**What is the difference between having and where clause?**

HAVING is used to specify a condition based on aggregate functions used in select after grouping.

The WHERE clause performs select before grouping. The HAVING clause performs select after grouping.

Unlike HAVING clause, the WHERE clause cannot contain aggregate functions.

**How to print duplicate rows in a table?**

| **Name** | **Section** |
| --- | --- |
| abc | CS1 |
| bcd | CS2 |
| abc | CS1 |

In the above table, we can find duplicate row using below query.

SELECT name, section

FROM tbl

GROUP BY name, section

HAVING COUNT(\*) > 1

**What is Join?**

Join is an operation that combines data from two or more tables by using cartesian product.

Natural Join combines data from two tables and only keeps the rows that have the same value on the common attributes.

Left Outer Join combines data from two tables based on the common attributes, but non-matching tuples from the left table are kept.

Right Outer Join combines data from two tables based on the common attributes, but non-matching tuples from the right table are kept.

Full Outer Join combines data from two tables based on the common attributes, but non-matching tuples from the both tables are kept.

**What is Identity?**

Identity (or AutoNumber) is a column that automatically generates numeric values. A start and increment value can be set, but most DBA leave these at 1. A GUID column also generates numbers; the value of this cannot be controlled. Identity/GUID columns do not need to be indexed.

**What is a view in SQL? How to create one**

A [view](http://en.wikipedia.org/wiki/View_(SQL)) is an intermediate table used in SQL queries. If the result of a certain query is used often, we can define it as a VIEW. We can create using create VIEW syntax.

CREATE VIEW view\_name(list) AS

SELECT column\_name\_list

FROM table\_name

WHERE condition

**What are the uses of view?**

1. Views can represent a subset of the data in a table. Therefore, it hides the complexity of the table or limits the access of the underlying tables to the outside world: a given user may have permission to query the view, while denied access to the rest of the base table.

2. Views can join and aggregate multiple tables into one single virtual table

3. Views take very little space to store; the database contains only the definition of the view, not the copy of all the data which it presents.  
  
  
**What is a Trigger?**

A trigger is a set of actions to be taken if the associated Insert, Update or Delete operations take place. The trigger is executed automatically whenever the associated operation is executed on a table. Triggers can be useful to maintain integrity in database.

**What is a stored procedure?**

A [stored procedure](http://en.wikipedia.org/wiki/Stored_procedure) is like a function that contains a set of operations organized together. It contains a set of operations that are commonly used in an application to do some common database tasks.

**What is the difference between Trigger and Stored Procedure?**

Unlike stored procedures, triggers cannot be called directly. They can only be associated with queries.

**What is a transaction? What are ACID properties?**

A [Database Transaction](http://en.wikipedia.org/wiki/Database_transaction) is a set of operations that must be treated as whole, means either all operations are executed or none of them.

[ACID](http://en.wikipedia.org/wiki/ACID) (Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee that database transactions are processed reliably.

**What are indexes?**   
A [database index](http://en.wikipedia.org/wiki/Database_index) is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and the use of more storage space to maintain the extra copy of data.

**What are clustered and non-clustered Indexes?**

Clustered indexes is the index according to which data is physically stored on disk. Therefore, only one clustered index can be created on a given database table.

Non-clustered indexes don’t define physical ordering of data, but logical ordering. Typically, a tree is created whose leaves point to disk records. [B-Tree](http://en.wikipedia.org/wiki/B-tree) or [B+ tree](http://en.wikipedia.org/wiki/B+_tree) are used for this purpose.