

# Database (DBMS) interview questions and answers are below

**Questions : What is database or database management systems (DBMS)? and - What's the difference between file and database? Can files qualify as a database?**  
1

**Answers :** Database provides a systematic and organized way of storing, managing and retrieving from collection of logically related information.  
1

Secondly the information has to be persistent, that means even after the application is closed the information should be persisted.

Finally it should provide an independent way of accessing data and should not be dependent on the application to access the information.

Main difference between a simple file and database that database has independent way (SQL) of accessing information while simple files do not. File meets the storing, managing and retrieving part of a database but not the independent way of accessing data. Many experienced programmers think that the main difference is that file can not provide multi-user capabilities which a DBMS provides. But if we look at some old COBOL and C programs where file was the only means of storing data, we can see functionalities like locking, multi-user etc provided very efficiently. So it's a matter of debate if some interviewers think this as a main difference between files and database accept it... going in to debate is probably losing a job.

**Questions : What is SQL ?**  
2

**Answers :** SQL stands for Structured Query Language. SQL is an ANSI (American National Standards Institute) standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database.  
2

**Questions : What's difference between DBMS and RDBMS ?**  
3

**Answers :** DBMS provides a systematic and organized way of storing, managing and retrieving from collection of logically related information. RDBMS also provides what DBMS provides but above that it provides relationship integrity. So in short we can say  
3

**RDBMS = DBMS + REFERENTIAL INTEGRITY**

These relations are defined by using "Foreign Keys" in any RDBMS. Many DBMS companies claimed their DBMS product was a RDBMS compliant, but according to industry rules and regulations if the DBMS fulfills the twelve CODD rules it's truly a RDBMS. Almost all DBMS (SQL SERVER, ORACLE etc) fulfill all the twelve CODD rules and are considered as truly RDBMS.

**Questions : What are CODD rules?**  
4

**Answers :** In 1969 Dr. E. F. Codd laid down some 12 rules which a DBMS should adhere in order to get the logo of a true RDBMS.  
4

**Rule 1: Information Rule.**

"All information in a relational data base is represented explicitly at the logical level and in exactly one way - by values in tables."

**Rule 2: Guaranteed access Rule.**

"Each and every datum (atomic value) in a relational data base is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name."

In flat files we have to parse and know exact location of field values. But if a DBMS is truly RDBMS you can access the value by specifying the table name, field name, for instance Customers.Fields ['Customer Name'].

**Rule 3: Systematic treatment of null values.**

"Null values (distinct from the empty character string or a string of blank characters and distinct from zero or any other number) are supported in fully relational DBMS for representing missing information and inapplicable information in a systematic way, independent of data type."

**Rule 4: Dynamic on-line catalog based on the relational model.**

"The data base description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data." The Data Dictionary is held within the RDBMS, thus there is no-need for off-line volumes to tell you the structure of the database.

**Rule 5: Comprehensive data sub-language Rule.**

"A relational system may support several languages and various modes of terminal use (for example, the fill-in-the-blanks mode). However, there must be at least one language whose statements are expressible, per some well-defined syntax, as character strings and that is comprehensive in supporting all the following items

Data Definition  
View Definition  
Data Manipulation (Interactive and by program).  
Integrity Constraints  
Authorization.  
Transaction boundaries ( Begin , commit and rollback)

**Rule 6: .View updating Rule**

"All views that are theoretically updatable are also updatable by the system."

**Rule 7: High-level insert, update and delete.**

"The capability of handling a base relation or a derived relation as a single operand applies not only to the retrieval of data but also to the insertion, update and deletion of data."

**Rule 8: Physical data independence.**

"Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representations or access methods."

**Rule 9: Logical data independence.**

"Application programs and terminal activities remain logically unimpaired when information-preserving changes of any kind that theoretically permit un-impairment are made to the base tables."

**Rule 10: Integrity independence.**

"Integrity constraints specific to a particular relational data base must be definable in the relational data sub-language and storable in the catalog, not in the application programs." **Rule**

**11: Distribution independence.**

"A relational DBMS has distribution independence."

**Rule 12: Non-subversion Rule.**

"If a relational system has a low-level (single-record-at-a-time) language, that low level cannot be used to subvert or bypass the integrity Rules and constraints expressed in the higher level relational language (multiple-records-at-a-time)."

**Questions : 5** What are E-R diagrams?

**Answers :** E-R diagram also termed as Entity-Relationship diagram shows relationship between various tables in the database. .

**Questions : 6** How many types of relationship exist in database designing?

**Answers :** There are three major relationship models:-  
6 One-to-one  
One-to-many  
Many-to-many

**Questions : 7** 7.What is normalization? What are different type of normalization?

**Answers :** There is set of rules that has been established to aid in the design of tables that are meant to be connected through relationships. This set of rules is known as Normalization.

Benefits of Normalizing your database include:

=>Avoiding repetitive entries

=>Reducing required storage space

=>Preventing the need to restructure existing tables to accommodate new data.

=>Increased speed and flexibility of queries, sorts, and summaries.

**Following are the three normal forms :-**

**First Normal Form**

For a table to be in first normal form, data must be broken up into the smallest un possible. In addition to breaking data up into the smallest meaningful values, tables first normal form should not contain repetitions groups of fields.

**Second Normal form**

The second normal form states that each field in a multiple field primary keytable must be directly related to the entire primary key. Or in other words,each non-key field should be a fact about all the fields in the primary key.

**Third normal form**

A non-key field should not depend on other Non-key field.

**Questions : 8** What is denormalization ?

**Answers :** Denormalization is the process of putting one fact in numerous places (its vice-versa of normalization).Only one valid reason exists for denormalizing a relational design - to enhance performance.The sacrifice to performance is that you increase redundancy in database.

**Questions : 9** Can you explain Fourth Normal Form and Fifth Normal Form ?

**Answers : 9** In fourth normal form it should not contain two or more independent multi-valued attributes about an entity and it should satisfy "Third Normal form".  
Fifth normal form deals with reconstructing information from smaller pieces of information. These smaller pieces of information can be maintained with less redundancy.

**Questions : 10** Have you heard about sixth normal form?

**Answers : 10** If we want relational system in conjunction with time we use sixth normal form. At this moment SQL Server does not support it directly.

**Questions : 11** What are DML and DDL statements?

**Answers : 11** DML stands for Data Manipulation Statements. They update data values in table. Below are the most important DDL statements:-  
=>SELECT - gets data from a database table  
=> UPDATE - updates data in a table  
=> DELETE - deletes data from a database table  
=> INSERT INTO - inserts new data into a database table  
  
DDL stands for Data definition Language. They change structure of the database objects like table, index etc. Most important DDL statements are as shown below: -  
=>CREATE TABLE - creates a new table in the database.  
=>ALTER TABLE - changes table structure in database.  
=>DROP TABLE - deletes a table from database  
=> CREATE INDEX - creates an index  
=> DROP INDEX - deletes an index

**Questions : 12** How do we select distinct values from a table?

**Answers : 12** DISTINCT keyword is used to return only distinct values. Below is syntax: - Column age and Table pcdsEmp  
SELECT DISTINCT age FROM pcdsEmp

**Questions : 13** What is Like operator for and what are wild cards?

**Answers : 13** LIKE operator is used to match patterns. A "%" sign is used to define the pattern.  
Below SQL statement will return all words with letter "S"  
SELECT \* FROM pcdsEmployee WHERE EmpName LIKE 'S%'  
Below SQL statement will return all words which end with letter "S"  
SELECT \* FROM pcdsEmployee WHERE EmpName LIKE '%S'  
Below SQL statement will return all words having letter "S" in between  
SELECT \* FROM pcdsEmployee WHERE EmpName LIKE '%S%'  
"\_" operator (we can read as "Underscore Operator"). "\_" operator is the character defined at that point. In the below sample fired a query Select name from pcdsEmployee where name like '\_s%' So all name where second letter is "s" is returned.

**Questions : 14** Can you explain Insert, Update and Delete query?

**Answers : 14** Insert statement is used to insert new rows in to table. Update to update existing data in the table. Delete statement to delete a record from the table. Below code snippet for Insert, Update and Delete :-  
  
INSERT INTO pcdsEmployee SET name='rohit',age='24';  
UPDATE pcdsEmployee SET age='25' where name='rohit';  
DELETE FROM pcdsEmployee WHERE name = 'sonia';

**Questions : 15** What is order by clause?

**Answers : 15** ORDER BY clause helps to sort the data in either ascending order to descending order.  
Ascending order sort query  
SELECT name,age FROM pcdsEmployee ORDER BY age ASC  
Descending order sort query

SELECT name FROM pcdsEmployee ORDER BY age DESC

**Questions : 16** What is the SQL " IN " clause?

**Answers : 16** SQL IN operator is used to see if the value exists in a group of values. For instance the below SQL checks if the Name is either 'rohit' or 'Anuradha' SELECT \* FROM pcdsEmployee WHERE name IN ('Rohit','Anuradha') Also you can specify a not clause with the same. SELECT \* FROM pcdsEmployee WHERE age NOT IN (17,16)

**Questions : 17** Can you explain the between clause?

**Answers : 17** Below SQL selects employees born between '01/01/1975' AND '01/01/1978' as per mysql  
SELECT \* FROM pcdsEmployee WHERE DOB BETWEEN '1975-01-01' AND '2011-09-28'

**Questions : 18** we have an employee salary table how do we find the second highest from it?

**Answers : 18** below Sql Query find the second highest salary  
SELECT \* FROM pcdsEmployeeSalary a WHERE (2=(SELECT COUNT(DISTINCT(b.salary)) FROM pcdsEmployeeSalary b WHERE b.salary>=a.salary))

**Questions : 19** What are different types of joins in SQL?

**Answers : 19** **INNER JOIN**

Inner join shows matches only when they exist in both tables. Example in the below SQL there are two tables Customers and Orders and the inner join is made on Customers.Customerid and Orders.Customerid. So this SQL will only give you result with customers who have orders. If the customer does not have order it will not display that record.  
SELECT Customers.\*, Orders.\* FROM Customers INNER JOIN Orders ON Customers.CustomerID =Orders.CustomerID

#### **LEFT OUTER JOIN**

Left join will display all records in left table of the SQL statement. In SQL below customers with or without orders will be displayed. Order data for customers without orders appears as NULL values. For example, you want to determine the amount ordered by each customer and you need to see who has not ordered anything as well. You can also see the LEFT OUTER JOIN as a mirror image of the RIGHT OUTER JOIN (Is covered in the next section) if you switch the side of each table.

SELECT Customers.\*, Orders.\* FROM Customers LEFT OUTER JOIN Orders ON Customers.CustomerID =Orders.CustomerID

#### **RIGHT OUTER JOIN**

Right join will display all records in right table of the SQL statement. In SQL below all orders with or without matching customer records will be displayed. Customer data for orders without customers appears as NULL values. For example, you want to determine if there are any orders in the data with undefined CustomerID values (say, after a conversion or something like it). You can also see the RIGHT OUTER JOIN as a mirror image of the LEFT OUTER JOIN if you switch the side of each table.

SELECT Customers.\*, Orders.\* FROM Customers RIGHT OUTER JOIN Orders ON Customers.CustomerID =Orders.CustomerID

**Questions : 20** What is "CROSS JOIN"? or What is Cartesian product?

**Answers : 20** "CROSS JOIN" or "CARTESIAN PRODUCT" combines all rows from both tables. Number of rows will be product of the number of rows in each table. In real life scenario I can not imagine where we will want to use a Cartesian product. But there are scenarios where we would like permutation and combination probably Cartesian would be the easiest way to achieve it.

**Questions : 21** How to select the first record in a given set of rows?

**Answers : 21** Select top 1 \* from sales.salesperson

**Questions : 22** What is the default "-SORT " order for a SQL?

**Answers :** ASCENDING  
**22**

**Questions : 23** **What is a self-join?**

**Answers :** If we want to join two instances of the same table we can use self-join.  
**23**

**Questions : 24** **What's the difference between DELETE and TRUNCATE ?**

**Answers :** Following are difference between them:  
**24** =>>DELETE TABLE syntax logs the deletes thus making the delete operations low. TRUNCATE table does not log any information but it logs information about deallocation of data page of the table. So TRUNCATE table is faster as compared to delete table.  
=>>DELETE table can have criteria while TRUNCATE can not.  
=>> TRUNCATE table can not have triggers.

**Questions : 25** **What's the difference between "UNION" and "UNION ALL" ?**

**Answers :** UNION SQL syntax is used to select information from two tables. But it selects only distinct records from both the table. , while UNION ALL selects all records from both the tables.  
**25**

**Questions : 26** **What are cursors and what are the situations you will use them?**

**Answers :** SQL statements are good for set at a time operation. So it is good at handling set of data. But there are scenarios where we want to update row depending on certain criteria. we will loop through all rows and update data accordingly. There's where cursors come in to picture.  
**26**

**Questions : 27** **What is " Group by " clause?**

**Answers :** "Group by" clause group similar data so that aggregate values can be derived.  
**27**

**Questions : 28** **What is the difference between "HAVING" and "WHERE" clause?**

**Answers :** "HAVING" clause is used to specify filtering criteria for "GROUP BY", while "WHERE" clause applies on normal SQL.  
**28**

**Questions : 29** **What is a Sub-Query?**

**Answers :** A query nested inside a SELECT statement is known as a subquery and is an alternative to complex join statements. A subquery combines data from multiple tables and returns results that are inserted into the WHERE condition of the main query. A subquery is always enclosed within parentheses and returns a column. A subquery can also be referred to as an inner query and the main query as an outer query. JOIN gives better performance than a subquery when you have to check for the existence of records.  
For example, to retrieve all EmployeeID and CustomerID records from the ORDERS table that have the EmployeeID greater than the average of the EmployeeID field, you can create a nested query, as shown:  
SELECT DISTINCT EmployeeID, CustomerID FROM ORDERS WHERE EmployeeID > (SELECT AVG(EmployeeID) FROM ORDERS)  
**29**

**Questions : 30** **What are Aggregate and Scalar Functions?**

**Answers :** Aggregate and Scalar functions are in built function for counting and calculations. Aggregate functions operate against a group of values but returns only one value.  
**30** AVG(column) :- Returns the average value of a column  
COUNT(column) :- Returns the number of rows (without a NULL value) of a column  
COUNT(\*) :- Returns the number of selected rows  
MAX(column) :- Returns the highest value of a column  
MIN(column) :- Returns the lowest value of a column  
Scalar functions operate against a single value and return value on basis of the single value.  
UCASE(c) :- Converts a field to upper case

LCASE(c) :- Converts a field to lower case  
MID(c,start[,end]) :- Extract characters from a text field  
LEN(c) :- Returns the length of a text

**Questions : 31 Can you explain the SELECT INTO Statement?**

**Answers : 31** SELECT INTO statement is used mostly to create backups. The below SQL backup the Employee table in to the EmployeeBackup table. One point to be noted is that the structure of pcdsEmployeeBackup and pcdsEmployee table should be same. `SELECT * INTO pcdsEmployeeBackup FROM pcdsEmployee`

**Questions : 32 What is a View?**

**Answers : 32** View is a virtual table which is created on the basis of the result set returned by the select statement.  
`CREATE VIEW [MyView] AS SELECT * from pcdsEmployee where LastName = 'singh'`  
In order to query the view  
`SELECT * FROM [MyView]`

**Questions : 33 What is SQL injection ?**

**Answers : 33** It is a Form of attack on a database-driven Web site in which the attacker executes unauthorized SQL commands by taking advantage of insecure code on a system connected to the Internet, bypassing the firewall. SQL injection attacks are used to steal information from a database from which the data would normally not be available and/or to gain access to an organization's host computers through the computer that is hosting the database. SQL injection attacks typically are easy to avoid by ensuring that a system has strong input validation.  
As name suggest we inject SQL which can be relatively dangerous for the database. Example this is a simple SQL  
`SELECT email, passwd, login_id, full_name  
FROM members WHERE email = 'x'`  
Now somebody does not put "x" as the input but puts "x ; DROP TABLE members;".  
So the actual SQL which will execute is :-  
`SELECT email, passwd, login_id, full_name FROM members WHERE email = 'x' ; DROP TABLE members;`  
Think what will happen to your database.

**Questions : 34 What is Data Warehousing ?**

**Answers : 34** **Data Warehousing** is a process in which the data is stored and accessed from central location and is meant to support some strategic decisions. Data Warehousing is not a requirement for Data mining. But just makes your Data mining process more efficient.  
  
Data warehouse is a collection of integrated, subject-oriented databases designed to support the decision-support functions (DSF), where each unit of data is relevant to some moment in time.

**Questions : 35 What are Data Marts?**

**Answers : 35** Data Marts are smaller section of Data Warehouses. They help data warehouses collect data. For example your company has lot of branches which are spanned across the globe. Head-office of the company decides to collect data from all these branches for anticipating market. So to achieve this IT department can setup data mart in all branch offices and a central data warehouse where all data will finally reside.

**Questions : 36 What are Fact tables and Dimension Tables ? What is Dimensional Modeling and Star Schema Design**

**Answers : 36** When we design transactional database we always think in terms of normalizing design to its least form. But when it comes to designing for Data warehouse we think more in terms of denormalizing the database. Data warehousing databases are designed using Dimensional Modeling. Dimensional Modeling uses the existing relational database structure and builds on that.  
There are two basic tables in dimensional modeling: -  
**Fact Tables.**  
**Dimension Tables.**  
Fact tables are central tables in data warehousing. Fact tables have the actual aggregate values

which will be needed in a business process. While dimension tables revolve around fact tables. They describe the attributes of the fact tables.

**Questions : 37** What is Snow Flake Schema design in database? What's the difference between Star and Snow flake schema?

**Answers : 37** Star schema is good when you do not have big tables in data warehousing. But when tables start becoming really huge it is better to denormalize. When you denormalize star schema it is nothing but snow flake design. For instance below customeraddress table is been normalized and is a child table of Customer table. Same holds true for Salesperson table.

**Questions : 38** What is ETL process in Data warehousing? What are the different stages in "Data warehousing"?

**Answers : 38** ETL (Extraction, Transformation and Loading) are different stages in Data warehousing. Like when we do software development we follow different stages like requirement gathering, designing, coding and testing. In the similar fashion we have for data warehousing.

**Extraction:-**

In this process we extract data from the source. In actual scenarios data source can be in many forms EXCEL, ACCESS, Delimited text, CSV (Comma Separated Files) etc. So extraction process handle's the complexity of understanding the data source and loading it in a structure of data warehouse.

**Transformation:-**

This process can also be called as cleaning up process. It's not necessary that after the extraction process data is clean and valid. For instance all the financial figures have NULL values but you want it to be ZERO for better analysis. So you can have some kind of stored procedure which runs through all extracted records and sets the value to zero.

**Loading:-**

After transformation you are ready to load the information in to your final data warehouse database.

**Questions : 39** What is Data mining ?

**Answers : 39** **Data mining** is a concept by which we can analyze the current data from different perspectives and summarize the information in more useful manner. It's mostly used either to derive some valuable information from the existing data or to predict sales to increase customer market. There are two basic aims of Data mining:-

**Prediction: -**

From the given data we can focus on how the customer or market will perform. For instance we are having a sale of 40000 \$ per month in India, if the same product is to be sold with a discount how much sales can the company expect.

**Summarization: -**

To derive important information to analyze the current business scenario. For example a weekly sales report will give a picture to the top management how we are performing on a weekly basis?

**Questions : 40** Compare Data mining and Data Warehousing ?

**Answers : 40** "Data Warehousing" is technical process where we are making our data centralized while "Data mining" is more of business activity which will analyze how good your business is doing or predict how it will do in the future coming times using the current data. As said before "Data Warehousing" is not a need for "Data mining". It's good if you are doing "Data mining" on a "Data Warehouse" rather than on an actual production database. "Data Warehousing" is essential when we want to consolidate data from different sources, so it's like a cleaner and matured data which sits in between the various data sources and brings then in to one format. "Data Warehouses" are normally physical entities which are meant to improve accuracy of "Data mining" process. For example you have 10 companies sending data in different format, so you create one physical database for consolidating all the data from different company sources, while "Data mining" can be a physical model or logical model. You can create a database in "Data mining" which gives you reports of net sales for this year for all companies. This need not be a physical database as such but a simple query.

**Questions : 41** What are indexes? What are B-Trees?

**Answers : 41** Index makes your search faster. So defining indexes to your database will make your search faster. Most of the indexing fundamentals use "B-Tree" or "Balanced-Tree" principle. It's not a principle that is something is created by SQL Server or ORACLE but is a mathematical derived fundamental. In order that "B-tree" fundamental work properly both of the sides should be

balanced.

**Questions : 42** I have a table which has lot of inserts, is it a good database design to create indexes on that table?  
**Insert's are slower on tables which have indexes, justify it?or Why do page splitting happen?**

**Answers : 42** All indexing fundamentals in database use "B-tree" fundamental. Now whenever there is new data inserted or deleted the tree tries to become unbalance.  
Creates a new page to balance the tree.  
Shuffle and move the data to pages.  
So if your table is having heavy inserts that means it's transactional, then you can visualize the amount of splits it will be doing. This will not only increase insert time but will also upset the end-user who is sitting on the screen. So when you forecast that a table has lot of inserts it's not a good idea to create indexes.

**Questions : 43** What are the two types of indexes and explain them in detail? or What's the difference between clustered and non-clustered indexes?

**Answers : 43** There are basically two types of indexes: -  
Clustered Indexes.  
Non-Clustered Indexes.  
In clustered index the non-leaf level actually points to the actual data. In Non-Clustered index the leaf nodes point to pointers (they are rowid's) which then point to actual data.