SQL questions

1. What is the basic syntax for SQL statements

SELECT columns  
FROM tables  
WHERE predicates;

1. For example there is a table called Customers.

The table has following columns

Product id, Product name, City, state

|  |  |  |  |
| --- | --- | --- | --- |
| Product id | Product name | City | state |
| 1 | Tv | Atlanta | GA |
| 2 | phone | boston | MA |
| 3 | car | New york | NY |
| 4 | furniture | New jersey | CA |

I want to see data only for a particular city like atlanta. How do I do this

***Answer:***

Select \* from customers

Where city =

1. I want to see all the cities which start with N

Answer:

Select \* from customers

Where city like ‘N%’

1. I want to see the data for multiple cities

Answer:

Select \* from customers

Where city **in** (‘’,’’)

1. Group by

SELECT department, SUM(sales) as "Total sales"  
FROM order\_details  
GROUP BY department;

1. When can you use having clause

The HAVING clause is used in combination with the GROUP BY clause

1. What are the different types of joins

Inner join – matching values from both the tables

Outer join –

Left outer join – all the values from the left table and matching rows from right table

Right outer join – all the values from the right table and matching rows from left table

Full outer join – all the values from both the tables

1. What is a sub query

Query within a query

1. In union query what are the conditions to be met to define union query

In UNION query both the tables must have the same number of fields and similar data types

1. What is the difference between union and union all

Union - It removes duplicate rows between the various "select" statements

Union all – returns duplicate values too

1. What does UPDATE statement do

The UPDATE statement allows you to update a single record or multiple records in a table.

The syntax for the UPDATE statement is:

UPDATE table  
SET column = expression  
WHERE predicates;

1. What does INSERT statement do

The INSERT statement allows you to insert a single record or multiple records into a table.

The syntax for the INSERT statement is:

INSERT INTO table  
(column-1, column-2, ... column-n)  
VALUES  
(value-1, value-2, ... value-n);

1. The DELETE statement allows you to delete a single record or multiple records from a table.

The syntax for the DELETE statement is:

DELETE FROM table  
WHERE predicates;

1. Create table

The CREATE TABLE statement allows you to create and define a table.

The basic syntax for a CREATE TABLE statement is:

CREATE TABLE table\_name  
( column1 datatype null/not null,  
column2 datatype null/not null,  
...  
);

1. Alter table

ALTER TABLE statement allows you to rename an existing table. It can also be used to add, modify, or drop a column from an existing table.

ALTER TABLE table\_name  
ADD column\_name column-definition;

1. **SQL: VIEWS**

A view is, in essence, a virtual table. It does not physically exist. Rather, it is created by a query joining one or more tables.

**Creating a VIEW**

The syntax for creating a VIEW is:

CREATE VIEW view\_name AS  
SELECT columns  
FROM table  
WHERE predicates;

1. select  
   CASE  
   WHEN a < b THEN 'hello'  
   WHEN d < e THEN 'goodbye'  
   END  
   from suppliers;
2. what is concat function

In Oracle/PLSQL, the **concat** function allows you to concatenate two strings together.

1. Substring

In Oracle/PLSQL, the **substr** functions allows you to extract a substring from a string.

1. To\_char

In Oracle/PLSQL, the **to\_char** function converts a number or date to a string.

**Difference between dates**

select datediff (dd,'2-06-2007','7-06-2007')

**How do you add a column to a table?**

**A. ALTER TABLE Department**

**ADD (AGE, NUMBER);**

**How do you change value of the field?**

**A. UPDATE EMP\_table**

**set number = 200 where item\_munber = â€˜CDâ€™;**

concatenation expression

firstname||’ ‘||lastname

**The COUNT function tells you how many rows were in the result set.**

**SELECT COUNT(\*) FROM TESTING.QA**

**2) The AVG function tells you the average value of a numeric column.**

**SELECT MAX(SALARY) FROM TESTING.QA**

**3) The MAX and MIN functions tell you the maximum and minimum value of a numeric column.**

**SELECT MIN(SALARY) FROM TESTING.QA**

**4) The SUM function tells you the sum value of a numeric column.**

**SELECT SUM(SALARY) FROM TESTING.QA**

**Distinct**

SELECT DISTINCT columns  
FROM tables  
WHERE predicates;

SELECT \*  
FROM suppliers  
WHERE supplier\_name in ( 'IBM', 'Hewlett Packard', 'Microsoft');

SELECT \*  
FROM suppliers  
WHERE supplier\_id between 5000 AND 5010;

SELECT \*  
FROM orders  
WHERE order\_date between to\_date ('2003/01/01', 'yyyy/mm/dd')  
AND to\_date ('2003/12/31', 'yyyy/mm/dd');

SELECT \*  
FROM suppliers  
WHERE supplier\_id not between 5000 and 5500;

SELECT \*  
FROM suppliers  
WHERE EXISTS  
(select \*  
from orders  
where suppliers.supplier\_id = orders.supplier\_id);

SELECT supplier\_city  
FROM suppliers  
WHERE supplier\_name = 'IBM'  
ORDER BY supplier\_city DESC;

select suppliers.name, subquery1.total\_amt  
from suppliers,  
(select supplier\_id, Sum(orders.amount) as total\_amt  
from orders  
group by supplier\_id) subquery1,  
where subquery1.supplier\_id = suppliers.supplier\_id;

select supplier\_id  
from suppliers  
MINUS  
select supplier\_id  
from orders;

UPDATE suppliers  
SET city = 'Santa Clara'  
WHERE supplier\_name = 'NVIDIA';

INSERT INTO suppliers  
(supplier\_id, supplier\_name)  
VALUES  
(24553, 'IBM');

DELETE FROM suppliers  
WHERE supplier\_name = 'IBM';

|  |  |  |
| --- | --- | --- |
| ALTER TABLE supplier | | |
| ADD ( | supplier\_name | varchar2(50), |
|  | city | varchar2(45) ); |

DROP TABLE supplier;

length('Tech on the Net')

lower('Tech on the Net')

replace('123123tech', '123');

|  |  |
| --- | --- |
| rtrim('tech '); | would return 'tech' |
| substr('This is a test', 6, 2) | would return 'is' |

|  |  |
| --- | --- |
| trim(' tech ') | would return 'tech' |

|  |  |
| --- | --- |
| upper('Tech on the Net'); | would return 'TECH ON THE NET' |

**How would you find out the total number of rows in a table?**

Use SELECT COUNT(\*) ... in query   
**How do you eliminate duplicate values in SELECT ?**

Use SELECT DISTINCT ... in SQL query

How you insert records into a table Using SQL INSERT statement   
**How do you delete record from a table ?**

Using DELETE statement

Example : DELETE FROM EMP

**How do you select a row using indexes?**

Specify the indexed columns in the WHERE clause of query.

**How do you find the maximum value in a column?**

Use SELECT MAX(...) .. in query

**How do you retrieve the first 5 characters of FIRSTNAME column of table EMP ?**

SELECT SUBSTR(FIRSTNAME,1,5) FROM EMP

**My SQL statement SELECT AVG(SALARY) FROM EMP yields inaccurate results. Why?**

Because SALARY is not declared to have NULLs and the employees for whom the  
salary is not known are also counted.

**How do you concatenate the FIRSTNAME and LASTNAME from EMP table to give a complete name?**

SELECT FIRSTNAME || ‘ ‘ || LASTNAME FROM EMP

**What is UNION,UNION ALL in SQL?**

UNION : eliminates duplicates

UNION ALL: retains duplicates

Both these are used to combine the results of different SELECT statements.

Suppose I have five SQL SELECT statements connected by UNION/UNION ALL, how many times  
**should I specify UNION to eliminate the duplicate rows?**

Once.

**In the WHERE clause what is BETWEEN and IN?**

BETWEEN supplies a range of values while IN supplies a list of values.

**Is BETWEEN inclusive of the range values specified?**

Yes.

**What is 'LIKE' used for in WHERE clause? What are the wildcard characters?**

LIKE is used for partial string matches. ‘%’ ( for a string of any character )

and ‘\_’ (for any single character ) are the two wild card characters.

**When do you use a LIKE statement?**

To do partial search e.g. to search employee by name, you need not specify

the complete name; using LIKE, you can search for partial string matches.

Example SQL : SELECT EMPNO FROM EMP

WHERE EMPNAME LIKE 'RAMESH%'

% is used to represent remaining all characters in the name.

This query fetches all records contains RAMESH in six characters.

**What do you accomplish by GROUP BY ... HAVING clause?**

GROUP BY partitions the selected rows on the distinct values of the column on

which you group by. HAVING selects GROUPs which match the criteria specified

**Consider the employee table with column PROJECT nullable. How can you get a list  
of employees who are not assigned to any project?**

SQL: SELECT EMPNO

FROM EMP

WHERE PROJECT IS null;

**What are the large objects supported by oracle and db2?**

Blob, Clob (Binary Large Objects, Character Large Objects)

**What's the difference between a primary key and a unique key?**

Primary key won’t allow nulls, unique key allow nulls.

Both Primary key and Unique key enforce the uniqueness of the column on which they are defined.

**What is a unique constraint?**

A unique constraint is a single field or combination of fields that uniquely defines a record. Some of the fields can contain null values as long as the combination of values is unique.

**What is a check constraint?**

A check constraint allows you to specify a condition on each row in a table.

**What is an Index?**

An index is a performance-tuning method of allowing faster retrieval of records. An index creates an entry for each value that appears in the indexed columns. By default, Oracle creates B-tree indexes.

**Oracle/PLSQL: Creating Functions**

In Oracle, you can create your own functions.

The syntax for a function is:

CREATE [OR REPLACE] FUNCTION function\_name

[ (parameter [,parameter]) ]

RETURN return\_datatype

IS | AS

[declaration\_section]

BEGIN

executable\_section

[EXCEPTION

exception\_section]

END [function\_name];

**What is a join and explain different types of joins?**

INNER JOIN

OUTER JOIN

LEFT OUTER JOIN

RIGHT OUTER JOIN

FULL OUTER JOIN

INNER JOIN

**What is a self join?**

Joining two instances of a same table.

Sample SQL : SELECT A.EMPNAME , B.EMPNAME

FROM EMP A, EMP B

WHERE A.MGRID = B.EMPID

**What is a transaction and ACID?**

Transaction - A transaction is a logicl unint of work. All steps must be commited or rolled back.

ACID - Atomicity, Consistency, Isolation and Duralbility, these are properties of a transaction.

**Materialized Query Tables in db2 ( This feature might not be available in oracle) ?**

Materialized Query Tables or MQTs are also known as automatic summary

tables. A materialized query table (MQT) is a table whose definition is based upon the result of a

query. The data that is contained in an MQT is derived from one or more tables on which the materialized

query table definition is based. MQT improve the query performance.

Sample SQL to creat MQT.

CREATE TABLE CUSTOMER\_ORDER AS

(SELECT SUM(AMOUNT) AS TOTAL\_SUM,

TRANS\_DT,

STATUS

FROM DB2INST2.CUSTOMER\_ORDER

WHERE TRANS\_DT BETWEEN '1/1/2001' AND '12/31/2001'

GROUP BY TRANS\_DT,

STATUS)

DATA INITIALLY DEFERRED REFRESH DEFERRED;

**Software testing - Questions and Answers - SQL Interview Questions**

**1. Q. What does SQL stand for?**

**A. Structured Query Language**

**Q. Who was E. F. Codd?**

**A. He was the original inventor of the relational model.**

**2. Q. How do you select all records from the table?**

**A. Select \* from table\_name;**

**Q. What do you understand by the term referential integrity?**

**Referential integrity is a database concept that ensures that relationships between tables remain consistent. When one table has a foreign key to another table, the concept of referential integrity states that you may not add a record to the table that contains the foreign key unless there is a corresponding record in the linked table. It also includes the techniques known as cascading update and cascading delete, which ensure that changes made to the linked table are reflected in the primary table.**

**3. Q. What is a join?**

**A. Join is a process of retrieve pieces of data from different sets (tables) and returns them to the user or program as one â€œjoinedâ€ collection of data.**

**4. Q. What kinds of joins do you know? Give examples.**

**A. We have self join, outer joint (LEFT, RIGHT), , cross-join ( Cartesian product n\*m rows returned)**

**Exp:**

**outer joint**

**SELECT Employee.Name, Department. DeptName**

**FROM Employee, Department**

**WHERE Employee.Employee\_ID = Department.Employee\_ID;**

**cross-join**

**SELECT \* FROM table1, table2;**

**self join**

**SELECT e1.name | |â€™ â€˜ | | e2.ename FROM emp e1, emp e2 WHERE e1. emp\_no = e2.emp\_no;**

**The following summarizes the result of the join operations:**

** The result of T1 INNER JOIN T2 consists of their paired rows where the**

**join-condition is true.**

** The result of T1 LEFT OUTER JOIN T2 consists of their paired rows where**

**the join-condition is true and, for each unpaired row of T1, the**

**concatenation of that row with the null row of T2. All columns derived**

**from T2 allow null values.**

** The result of T1 RIGHT OUTER JOIN T2 consists of their paired rows**

**where the join-condition is true and, for each unpaired row of T2, the**

**concatenation of that row with the null row of T1. All columns derived**

**from T1 allow null values.**

** The result of T1 FULL OUTER JOIN T2 consists of their paired rows and,**

**for each unpaired row of T2, the concatenation of that row with the null**

**row of T1 and, for each unpaired row of T1, the concatenation of that row**

**with the null row of T2. All columns derived from T1 and T2 allow null**

**values.**

**5. Q. How do you add record to a table?**

**A. INSERT into table\_name VALUES (â€˜ALEXâ€™ , 33 , â€˜Mâ€™);**

**6. Q. How do you add a column to a table?**

**A. ALTER TABLE Department**

**ADD (AGE, NUMBER);**

**7. Q. How do you change value of the field?**

**A. UPDATE EMP\_table**

**set number = 200 where item\_munber = â€˜CDâ€™;**

**update name\_table set status = 'enable' where phone = '4161112222';**

**update SERVICE\_table set REQUEST\_DATE = to\_date ('2006-03-04 09:29', 'yyyy-mm-dd hh24:MI') where phone = '4161112222';**

**8. Q. What does COMMIT do?**

**A. Saving all changes made by DML statements**

**9. Q. What is a primary key?**

**A. The column (columns) that has completely unique data throughout**

**the table is known as the primary key field.**

**10. Q. What are foreign keys?**

**A. Foreign key field â€“ is a field that links one table**

**to another tableâ€™s primary or foreign key.**

**11. Q. What is the main role of a primary key in a table?**

**A. The main role of a primary key in a data table is to maintain the internal integrity of a data table.**

**12. Q. Can a table have more than one foreign key defined?**

**A. A table can have any number of foreign keys defined. It can have only**

**one primary key defined.**

**13. Q. List all the possible values that can be stored in a BOOLEAN data field.**

**A. There are only two values that can be stored in a BOOLEAN data field:**

**-1(true) and 0(false).**

**14 Q. What is the highest value that can be stored in a BYTE data field?**

**A. The highest value that can be stored in a BYTE field is 255. or from -128**

**to 127. Byte is a set of Bits that represent a single character.**

**Usually there are 8 Bits in a Byte, sometimes more, depending on how**

**the measurement is being made. Each Char requires one byte of memory**

**and can have a value from 0 to 255 (or 0 to 11111111 in binary).**

**15. Q. How many places to the right of the decimal can be stored in a**

**CURRENCY data field?**

**A. The CURRENCY data type can store up to four places to the right of the**

**decimal. Any data beyond the fourth place will be truncated by Visual**

**Basic without reporting an error.**

**16. Q. What is a stored procedure?**

**A. A procedure is a group of PL/SQL statements that can be called by**

**a name. Procedures do not return values they perform tasks.**

**17. Q. Describe how NULLs work in SQL?**

**A. The NULL is how SQL handles missing values.**

**Arifthmetic operation with NULL in SQL will return a NULL.**

**18. Q. What is Normalization?**

**A. The process of table design is called normalization.**

**19. Q. What is referential integrity constraints?**

**A. Referential integrity constraints are rules**

**that are partnof the table in a database schema.**

**20. Q. What is Trigger?**

**A. Trigger will execute a block of procedural code**

**against the database when a table event occurs.**

**A2. A trigger defines a set of actions that are performed in response**

**to an insert, update, or delete operation on a specified table. When**

**such an SQL operation is executed, in this case the trigger has been**

**activated.**

**21. Q. Which of the following WHERE clauses will return only rows**

**that have a NULL in the PerDiemExpenses column?**

**A. WHERE PerDiemExpenses <>**

**B. WHERE PerDiemExpenses IS NULL**

**C. WHERE PerDiemExpenses = NULL**

**D. WHERE PerDiemExpenses NOT IN (\*)**

**A. B is correct ï¿½ When searching for a NULL value in a column, you must**

**use the keyword IS. No quotes are required around the keyword NULL.**

**22. Q. You issue the following query:SELECT FirstName FROM**

**StaffListWHERE FirstName LIKE'\_A%'Which names would be**

**returned by this query? Choose all that apply.**

**A. Allen**

**B. CLARK**

**C. JACKSON**

**D. David**

**A. C is correct ï¿½ Two wildcards are used with the LIKE operator.**

**The underscore (\_) stands for any one character of any**

**case, and the percent sign (%) stands for any number of**

**characters of any case including none. Because this string**

**starts with an underscore rather than a percent sign, it won't**

**return Allen or Clark because they represent zero and two**

**characters before the "A". If the LIKE string had been "%A%",**

**both of these values would have been returned.**

**David was not returned because all non-wild card characters**

**are case sensitive. Therefore, only strings**

**with an uppercase "A" as their second letter are returned**

**23. Q. Write a SQL SELECT query that only returns each city only once from Students table?**

**Do you need to order this list with an ORDER BY clause?**

**A. SELECT DISTINCT City**

**FROM Students;**

**The Distinct keyword automatically sorts all data**

**in ascending order. However, if you want the data**

**sorted in descending order, you have to use an ORDER BY clause**

**24. Q. Write a SQL SELECT sample of the concatenation operator.**

**A. SELECT LastName ||',' || FirstName, City FROM Students;**

**25. Q. How to rename column in the SQL SELECT query?**

**A. SELECT LastName ||',' || FirstName**

**AS "Student Name", City AS "Home City"**

**"FROM StudentsORDER BY "Student Name"**

**26. Q. Write SQL SELECT example how you limiting the rows returned with a WHERE clause.**

**A. SELECT InstructorID, Salary FROM Instructors**

**WHERE Salary > 5400 AND Salary < 6600;**

**27. Q. Write SQL SELECT query that returns the first and**

**last name of each instructor, the Salary,**

**and gives each of them a number.**

**A. SELECT FirstName, LastName, Salary,**

**ROWNUM FROM Instructors;**

**28. Q. Which of the following functions can be used only with numeric values?**

**(Choose all that apply.)**

**A. AVG**

**B. MIN**

**C. LENGTH**

**D. SUM**

**E. ROUND**

**A. A and D ï¿½ Only A and D are correct. The MIN function**

**works with any character, numeric, or date datatype.**

**The LENGTH function is a character function that returns**

**the number of letters in a character value. The ROUND**

**function works with both numeric and date values.**

**29. Q. Which function do you use to remove all padded characters**

**to the right of a character value in a column with a char datatype?**

**A. RTRIM**

**B. RPAD**

**C. TRIM**

**A. C ï¿½ The TRIM function is used to remove padded spaces.**

**LTRIM and RTRIM functions were included in earlier versions**

**of Oracle, but Oracle 8i has replaced them with a single**

**TRIM function**

**30. Q. Which statement do you use to eliminate padded spaces**

**between the month and day values in a function TO\_CHAR(SYSDATE,'Month, DD, YYYY') ?**

**A. To remove padded spaces, you use the "fm"**

**prefix before the date element that contains the spaces.**

**TO\_CHAR(SYSDATE,'fmMonth DD, YYYY')**

**31. Q. Is the WHERE clause must appear always before the GROUP BY clause in SQL SELECT ?**

**A. Yes.**

**The proper order for SQL SELECT**

**clauses is: SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY.**

**Only the SELECT and FROM clause are mandatory.**

**32. Q. How Oracle executes a statement with nested subqueries?**

**A. When Oracle executes a statement with nested subqueries,**

**it always executes the innermost query first. This query passes its**

**results to the next query and so on until it reaches the outermost query.**

**It is the outermost query that returns a result set.**

**33. Q. Which operator do you use to return all of the rows**

**from one query except rows are returned in a second query?**

**A. You use the MINUS operator to return all rows from one query except**

**where duplicate rows are found in a second query. The UNION operator**

**returns all rows from both queries minus duplicates. The UNION ALL operator**

**returns all rows from both queries including duplicates.**

**The INTERSECT operator returns only those rows that exist in both queries.**

**34. Q. How you will create a column alias? (Oracle 8i)**

**A. The AS keyword is optional when specifying a column alias.**

**You must enclose the column alias in double quotes when the alias**

**contains a space or lowercase letters. If you specify an alias in l**

**owercase letters without double quotes, the alias will appear in uppercase.**

**35 Q. Which of the following statements are Data Manipulation Language commands?**

**A. INSERT**

**B. UPDATE**

**C. GRANT**

**D. TRUNCATE**

**E. CREATE**

**A. A and B ï¿½ The INSERT and UPDATE statements are**

**Data Manipulation Language (DML) commands.**

**GRANT is a Data Control Language (DCL) command.**

**TRUNCATE and CREATE are Data Definition Language (DDL) commands**

**36. Question. What is Oracle locking?**

**A. Oracle uses locking mechanisms to protect data from**

**being destroyed by concurrent transactions.**

**37. Question. What Oracle lock modes do you know?**

**A. Oracle has two lock modes: shared or exclusive.**

**Shared locks are set on database resources so that many transactions**

**can access the resource.**

**Exclusive locks are set on resources that ensure**

**one transaction has exclusive access to the database resource**

**38. Question. What is query optimization?**

**A. Query optimization is the part of the query**

**process in which the database system compares**

**different query strategies and chooses the one with**

**the least expected cost**

**39. Question. What are the main components of Database management systems software.**

**A. The database management system software includes**

**components for storage management, concurrency control, transaction**

**processing, database manipulation interface, database definition interface,**

**and database control interface.**

**40. Question. What are the main attributes of database management system?**

**A. A database management system is composed of five elements: computer hardware, software, data, people (users), and operations procedures.**

**41. Question. What is transaction?**

**A. A transaction is a collection of applications**

**code and database manipulation code bound into an indivisible unit of execution.**

**it consists from:**

**BEGIN-TRANSACTION Name**

**Code**

**END TRANSACTION Name**

**42. Question. What databases do you know?**

**Informix**

**DB2**

**SQL**

**Oracle**

**43. Question. Explain SQL SELECT example:**

**select j.FILE\_NUM**

**from DB\_name.job j, DB\_name.address a**

**where j.JOB\_TYPE ='C'**

**AND j.COMPANY\_NAME = 'TEST6'**

**AND j.OFFICE\_ID = '101'**

**AND j.ACTIVE\_IND = 'Y'**

**AND a.ADDRESS\_STATUS\_ID = 'H'**

**AND a.OFFICE\_ID = '101'**

**AND a.FILE\_NUM = j.FILE\_NUM order by j.FILE\_NUM;**

**Answer: j and a aliases for table names. this is outer joint select statament from two tables.**

**44. Q. Describe some Conversion Functions that you know**

**A. TO\_CHAR converts a number / date to a string.**

**TO\_DATE converts a string (representing a date) to a date.**

**TO\_NUMBER converts a character string containing digits to a numeric data type, it accepts one parameter which is a column value or a string literal**

**45. Q. Describe some Group Functions that you know**

**A. 1) The COUNT function tells you how many rows were in the result set.**

**SELECT COUNT(\*) FROM TESTING.QA**

**2) The AVG function tells you the average value of a numeric column.**

**SELECT MAX(SALARY) FROM TESTING.QA**

**3) The MAX and MIN functions tell you the maximum and minimum value of a numeric column.**

**SELECT MIN(SALARY) FROM TESTING.QA**

**4) The SUM function tells you the sum value of a numeric column.**

**SELECT SUM(SALARY) FROM TESTING.QA**

**46. Question. What does DML stand for?**

**A. DML is Data Manipulation Language statements. (SELECT)**

**47. Question. What does DDL stand for?**

**A. DDL is Data Definition Language statements. (CREATE)**

**48. Question. What does DCL stand for?**

**A. DCL is Data Control Language statements. (COMMIT)**

**49. Question: Describe SQL comments.**

**A. SQL comments are introduced by two consecutive hyphens**

**(--) and ended by the end of the line.**

**50. Q. In what sequence SQL statement are processed?**

**A. The clauses of the subselect are processed in the following sequence (DB2):**

**1. FROM clause**

**2. WHERE clause**

**3. GROUP BY clause**

**4. HAVING clause**

**5. SELECT clause**

**6. ORDER BY clause**

**7. FETCH FIRST clause**

**51. Q. Describe TO\_DATE function.**

**A. The TO\_DATE function returns a timestamp from a character string**

**that has been interpreted using a character template.**

**TO\_DATE is a synonym for TIMESTAMP\_FORMAT.**

**52. Question:**

**In the domain table we have status as a numeric value from 01 to 04 and we**

**have text definition of these values in the design document.**

**Write SQL query to see the result as a text definitions that is corresponded**

**to these values. (DB2)**

**A. select TB1.member\_id, TB1.bu\_id, TB1.program, TB2.num,**

**case TB1.status**

**when '01' then 'Auto renew'**

**when '02' then 'Expired'**

**when '03' then 'Sold'**

**when '04' then ‘Terminated’**

**else TB\_name.status**

**end**

**from DB\_name.TB\_name1 TB1,**

**DB\_name.TB\_name2 TB2**

**where**

**TB1.program in ('com', 'org')**

**and TB1.member\_role = '100'**

**order by TB1.member\_id**

**fetch first 30 rows only**

**53. Question:**

**What's the logical difference, if any, between the following SQL expressions?**

**SELECT COUNT ( \* ) FROM T**

**SELECT SUM ( 1 ) FROM T**

**A. They're the same unless table T is empty, in which case the first yields**

**a one-column, one-row table containing a zero and the second yields a one-column,**

**one-row table "containing a null."**

**What is normalization? Explain different levels of normalization?**

In the design of a relational database management system (RDBMS), the process of organizing data to minimize redundancy is called normalization. The goal of database normalization is to decompose relations with anomalies in order to produce smaller, well-structured relations. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them

**What is denormalization and when would you go for it?**

Denormalisation is the process of attempting to optimise the read performance of a database by adding redundant data or by grouping data. In some cases, denormalisation helps cover up the inefficiencies inherent in relational database software. A relational normalised database imposes a heavy access load over physical storage of data even if it is well tuned for high performance.

**How do you implement one-to-one, one-to-many and many-to-many relationships while designing tables?**

One-to-One relationship can be implemented as a single table and rarely as two tables with primary and foreign key relationships. One-to-Many relationships are implemented by splitting the data into two tables with primary key and foreign key relationships. Many-to-Many relationships are implemented using a junction table with the keys from both the tables forming the composite primary key of the junction table.

It will be a good idea to read up a database designing fundamentals text book.

**What's the difference between a primary key and a unique key?**

Both primary key and unique enforce uniqueness of the column on which they are defined. But by default primary key creates a clustered index on the column, where are unique creates a nonclustered index by default. Another major difference is that, primary key doesn't allow NULLs, but unique key allows one NULL only.

**What are user defined datatypes and when you should go for them?**

User defined datatypes let you extend the base SQL Server datatypes by providing a descriptive name, and format to the database. Take for example, in your database, there is a column called Flight\_Num which appears in many tables. In all these tables it should be varchar(8). In this case you could create a user defined datatype called Flight\_num\_type of varchar(8) and use it across all your tables.

See sp\_addtype, sp\_droptype in books online.

**What is bit datatype and what's the information that can be stored inside a bit column?**

Bit datatype is used to store Boolean information like 1 or 0 (true or false). Until SQL Server 6.5 bit datatype could hold either a 1 or 0 and there was no support for NULL. But from SQL Server 7.0 onwards, bit datatype can represent a third state, which is NULL.

**Define candidate key, alternate key, composite key.**

A candidate key is one that can identify each row of a table uniquely. Generally a candidate key becomes the primary key of the table. If the table has more than one candidate key, one of them will become the primary key, and the rest are called alternate keys.

A key formed by combining at least two or more columns is called composite key.

**What are defaults? Is there a column to which a default can't be bound?**

A default is a value that will be used by a column, if no value is supplied to that column while inserting data. IDENTITY columns and timestamp columns can't have defaults bound to them. See CREATE DEFUALT in books online.

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

A transaction is a logical unit of work in which, all the steps must be performed or none. ACID stands for Atomicity, Consistency, Isolation, Durability. These are the properties of a transaction. For more information and explanation of these properties, see SQL Server books online or any RDBMS fundamentals text book.

**Explain different isolation levels**

An isolation level determines the degree of isolation of data between concurrent transactions. The default SQL Server isolation level is Read Committed. Here are the other isolation levels (in the ascending order of isolation): Read Uncommitted, Read Committed, Repeatable Read, Serializable. See SQL Server books online for an explanation of the isolation levels. Be sure to read about SET TRANSACTION ISOLATION LEVEL, which lets you customize the isolation level at the connection level.

CREATE INDEX myIndex ON myTable(myColumn)

**What type of Index will get created after executing the above statement?**

Non-clustered index. Important thing to note: By default a clustered index gets created on the primary key, unless specified otherwise.

**What's the maximum size of a row?**

8060 bytes. Don't be surprised with questions like 'what is the maximum number of columns per table'. Check out SQL Server books online for the page titled: "Maximum Capacity Specifications".

Explain Active/Active and Active/Passive cluster configurations

Hopefully you have experience setting up cluster servers. But if you don't, at least be familiar with the way clustering works and the two clusterning configurations Active/Active and Active/Passive. SQL Server books online has enough information on this topic and there is a good white paper available on Microsoft site.

**Explain the architecture of SQL Server**

This is a very important question and you better be able to answer it if consider yourself a DBA. SQL Server books online is the best place to read about SQL Server architecture. Read up the chapter dedicated to SQL Server Architecture.

**What is lock escalation?**

Lock escalation is the process of converting a lot of low level locks (like row locks, page locks) into higher level locks (like table locks). Every lock is a memory structure too many locks would mean, more memory being occupied by locks. To prevent this from happening, SQL Server escalates the many fine-grain locks to fewer coarse-grain locks. Lock escalation threshold was definable in SQL Server 6.5, but from SQL Server 7.0 onwards it's dynamically managed by SQL Server.

**What's the difference between DELETE TABLE and TRUNCATE TABLE commands?**

DELETE TABLE is a logged operation, so the deletion of each row gets logged in the transaction log, which makes it slow. TRUNCATE TABLE also deletes all the rows in a table, but it won't log the deletion of each row, instead it logs the deallocation of the data pages of the table, which makes it faster. Of course, TRUNCATE TABLE can be rolled back.

**Explain the storage models of OLAP**

Check out MOLAP, ROLAP and HOLAP in SQL Server books online for more information.

What are the new features introduced in SQL Server 2000 (or the latest release of SQL Server at the time of your interview)? What changed between the previous version of SQL Server and the current version?

This question is generally asked to see how current is your knowledge. Generally there is a section in the beginning of the books online titled "What's New", which has all such information. Of course, reading just that is not enough, you should have tried those things to better answer the questions. Also check out the section titled "Backward Compatibility" in books online which talks about the changes that have taken place in the new version.

**What are constraints? Explain different types of constraints.**

Constraints enable the RDBMS enforce the integrity of the database automatically, without needing you to create triggers, rule or defaults.

Types of constraints: NOT NULL, CHECK, UNIQUE, PRIMARY KEY, FOREIGN KEY

For an explanation of these constraints see books online for the pages titled: "Constraints" and "CREATE TABLE", "ALTER TABLE"

**Whar is an index? What are the types of indexes? How many clustered indexes can be created on a table? I create a separate index on each column of a table. what are the advantages and disadvantages of this approach?**

Indexes in SQL Server are similar to the indexes in books. They help SQL Server retrieve the data quicker.

Indexes are of two types. Clustered indexes and non-clustered indexes. When you craete a clustered index on a table, all the rows in the table are stored in the order of the clustered index key. So, there can be only one clustered index per table. Non-clustered indexes have their own storage separate from the table data storage. Non-clustered indexes are stored as B-tree structures (so do clustered indexes), with the leaf level nodes having the index key and it's row locater. The row located could be the RID or the Clustered index key, depending up on the absence or presence of clustered index on the table.

If you create an index on each column of a table, it improves the query performance, as the query optimizer can choose from all the existing indexes to come up with an efficient execution plan. At the same t ime, data modification operations (such as INSERT, UPDATE, DELETE) will become slow, as every time data changes in the table, all the indexes need to be updated. Another disadvantage is that, indexes need disk space, the more indexes you have, more disk space is used.

**What is RAID and what are different types of RAID configurations?**

RAID stands for Redundant Array of Inexpensive Disks, used to provide fault tolerance to database servers. There are six RAID levels 0 through 5 offering different levels of performance, fault tolerance. MSDN has some information about RAID levels and for detailed information, check out the RAID advisory board's homepage

**What are the steps you will take to improve performance of a poor performing query?**

This is a very open ended question and there could be a lot of reasons behind the poor performance of a query. But some general issues that you could talk about would be: No indexes, table scans, missing or out of date statistics, blocking, excess recompilations of stored procedures, procedures and triggers without SET NOCOUNT ON, poorly written query with unnecessarily complicated joins, too much normalization, excess usage of cursors and temporary tables.

Some of the tools/ways that help you troubleshooting performance problems are: SET SHOWPLAN\_ALL ON, SET SHOWPLAN\_TEXT ON, SET STATISTICS IO ON, SQL Server Profiler, Windows NT /2000 Performance monitor, Graphical execution plan in Query Analyzer.

Download the white paper on performance tuning SQL Server from Microsoft web site. Don't forget to check out sql-server-performance.com

**What are the steps you will take, if you are tasked with securing an SQL Server?**

Again this is another open ended question. Here are some things you could talk about: Preferring NT authentication, using server, database and application roles to control access to the data, securing the physical database files using NTFS permissions, using an unguessable SA password, restricting physical access to the SQL Server, renaming the Administrator account on the SQL Server computer, disabling the Guest account, enabling auditing, using multiprotocol encryption, setting up SSL, setting up firewalls, isolating SQL Server from the web server etc.

Read the white paper on SQL Server security from Microsoft website. Also check out My SQL Server security best practices

**What is a deadlock and what is a live lock? How will you go about resolving deadlocks?**

Deadlock is a situation when two processes, each having a lock on one piece of data, attempt to acquire a lock on the other's piece. Each process would wait indefinitely for the other to release the lock, unless one of the user processes is terminated. SQL Server detects deadlocks and terminates one user's process.

A livelock is one, where a request for an exclusive lock is repeatedly denied because a series of overlapping shared locks keeps interfering. SQL Server detects the situation after four denials and refuses further shared locks. A livelock also occurs when read transactions monopolize a table or page, forcing a write transaction to wait indefinitely.

Check out SET DEADLOCK\_PRIORITY and "Minimizing Deadlocks" in SQL Server books online. Also check out the article Q169960 from Microsoft knowledge base.

**What is blocking and how would you troubleshoot it?**

Blocking happens when one connection from an application holds a lock and a second connection requires a conflicting lock type. This forces the second connection to wait, blocked on the first.

Read up the following topics in SQL Server books online: Understanding and avoiding blocking, Coding efficient transactions.

**Explain CREATE DATABASE syntax**

Many of us are used to craeting databases from the Enterprise Manager or by just issuing the command: CREATE DATABAE MyDB. But what if you have to create a database with two filegroups, one on drive C and the  
other on drive D with log on drive E with an initial size of 600 MB and with a growth factor of 15%? That's why being a DBA you should be familiar with the CREATE DATABASE syntax. Check out SQL Server books  
online for more information.

**How to restart SQL Server in single user mode? How to start SQL Server in minimal configuration mode?**

SQL Server can be started from command line, using the SQLSERVR.EXE. This EXE has some very important parameters with which a DBA should be familiar with. -m is used for starting SQL Server in single user mode  
and -f is used to start the SQL Server in minimal confuguration mode. Check out SQL Server books online for more parameters and their explanations.

**As a part of your job, what are the DBCC commands that you commonly use for database maintenance?**

DBCC CHECKDB, DBCC CHECKTABLE, DBCC CHECKCATALOG, DBCC CHECKALLOC, DBCC SHOWCONTIG, DBCC SHRINKDATABASE, DBCC SHRINKFILE etc. But there are a whole load of DBCC commands which are very useful for DBAs. Check out SQL Server books online for more information.

**What are statistics, under what circumstances they go out of date, how do you update them?**

Statistics determine the selectivity of the indexes. If an indexed column has unique values then the selectivity of that index is more, as opposed to an index with non-unique values. Query optimizer uses these indexes in determining whether to choose an index or not while executing a query.

Some situations under which you should update statistics:  
1) If there is significant change in the key values in the index  
2) If a large amount of data in an indexed column has been added, changed, or removed (that is, if the distribution of key values has changed), or the table has been truncated using the TRUNCATE TABLE statement and then repopulated  
3) Database is upgraded from a previous version

Look up SQL Server books online for the following commands: UPDATE  
STATISTICS, STATS\_DATE, DBCC SHOW\_STATISTICS, CREATE STATISTICS, DROP  
STATISTICS, sp\_autostats, sp\_createstats, sp\_updatestats

**What are the different ways of moving data/databases between servers and databases in SQL Server?**

There are lots of options available, you have to choose your option depending upon your requirements. Some of the options you have are: BACKUP/RESTORE, dettaching and attaching databases, replication, DTS, BCP, logshipping, INSERT...SELECT, SELECT...INTO, creating INSERT scripts to generate data.

**Explian different types of BACKUPs avaialabe in SQL Server? Given a particular scenario, how would you go about choosing a backup plan?**

Types of backups you can create in SQL Sever 7.0+ are Full database backup, differential database backup, transaction log backup, filegroup backup. Check out the BACKUP and RESTORE commands in SQL Server books online. Be prepared to write the commands in your interview. Books online also has information on detailed backup/restore architecture and when one should go for a particular kind of backup.

**What is database replicaion? What are the different types of replication you can set up in SQL Server?**

Replication is the process of copying/moving data between databases on the same or different servers. SQL Server supports the following types of replication scenarios:

\* Snapshot replication  
\* Transactional replication (with immediate updating subscribers, with queued updating subscribers)  
\* Merge replication

See SQL Server books online for indepth coverage on replication. Be prepared to explain how different replication agents function, what are the main system tables used in replication etc.

**How to determine the service pack currently installed on SQL Server?**

The global variable @@Version stores the build number of the sqlservr.exe, which is used to determine the service pack installed. To know more about this process visit SQL Server service packs and versions.

**What are cursors? Explain different types of cursors. What are the disadvantages of cursors? How can you avoid cursors?**

Cursors allow row-by-row prcessing of the resultsets.

Types of cursors: Static, Dynamic, Forward-only, Keyset-driven. See books online for more information.

Disadvantages of cursors: Each time you fetch a row from the cursor, it results in a network roundtrip, where as a normal SELECT query makes only one rowundtrip, however large the resultset is. Cursors are also costly because they require more resources and temporary storage (results in more IO operations). Furthere, there are restrictions on the SELECT statements that can be used with some types of cursors.

Most of the times, set based operations can be used instead of cursors. Here is an example:

If you have to give a flat hike to your employees using the following criteria:

Salary between 30000 and 40000 -- 5000 hike  
Salary between 40000 and 55000 -- 7000 hike  
Salary between 55000 and 65000 -- 9000 hike

In this situation many developers tend to use a cursor, determine each employee's salary and update his salary according to the above formula. But the same can be achieved by multiple update statements or can be combined in a single UPDATE statement as shown below:

UPDATE tbl\_emp SET salary =  
CASE WHEN salary BETWEEN 30000 AND 40000 THEN salary + 5000  
WHEN salary BETWEEN 40000 AND 55000 THEN salary + 7000  
WHEN salary BETWEEN 55000 AND 65000 THEN salary + 10000  
END

Another situation in which developers tend to use cursors: You need to call a stored procedure when a column in a particular row meets certain condition. You don't have to use cursors for this. This can be achieved using WHILE loop, as long as there is a unique key to identify each row. For examples of using WHILE loop for row by row processing, check out the 'My code library' section of my site or search for WHILE.

**Write down the general syntax for a SELECT statements covering all the options.**

Here's the basic syntax: (Also checkout SELECT in books online for advanced syntax).

SELECT select\_list  
[INTO new\_table\_]  
FROM table\_source  
[WHERE search\_condition]  
[GROUP BY group\_by\_\_expression]  
[HAVING search\_condition]  
[ORDER BY order\_\_expression [ASC | DESC] ]

**What is a join and explain different types of joins?**

Joins are used in queries to explain how different tables are related. Joins also let you select data from a table depending upon data from another table.

Types of joins: INNER JOINs, OUTER JOINs, CROSS JOINs. OUTER JOINs are further classified as LEFT OUTER JOINS, RIGHT OUTER JOINS and FULL OUTER JOINS.

For more information see pages from books online titled: "Join Fundamentals" and "Using Joins".

**Can you have a nested transaction?**

Yes, very much. Check out BEGIN TRAN, COMMIT, ROLLBACK, SAVE TRAN and @@TRANCOUNT

**What is an extended stored procedure? Can you instantiate a COM object by using T-SQL?**

An extended stored procedure is a function within a DLL (written in a programming language like C, C++ using Open Data Services (ODS) API) that can be called from T-SQL, just the way we call normal stored procedures using the EXEC statement. See books online to learn how to create extended stored procedures and how to add them to SQL Server.

Yes, you can instantiate a COM (written in languages like VB, VC++) object from T-SQL by using sp\_OACreate stored procedure. Also see books online for sp\_OAMethod, sp\_OAGetProperty, sp\_OASetProperty, sp\_OADestroy. For an example of creating a COM object in VB and calling it from T-SQL, see 'My code library' section of this site.

**What is the system function to get the current user's user id?**

USER\_ID(). Also check out other system functions like USER\_NAME(),  
SYSTEM\_USER, SESSION\_USER, CURRENT\_USER, USER, SUSER\_SID(), HOST\_NAME().

**What are triggers? How many triggers you can have on a table? How to invoke a trigger on demand?**

Triggers are special kind of stored procedures that get executed automatically when an INSERT, UPDATE or DELETE operation takes place on a table.

In SQL Server 6.5 you could define only 3 triggers per table, one for INSERT, one for UPDATE and one for DELETE. From SQL Server 7.0 onwards, this restriction is gone, and you could create multiple triggers per each action. But in 7.0 there's no way to control the order in which the triggers fire. In SQL Server 2000 you could specify which trigger fires first or fires last using sp\_settriggerorder

Triggers can't be invoked on demand. They get triggered only when an associated action (INSERT, UPDATE, DELETE) happens on the table on which they are defined.

Triggers are generally used to implement business rules, auditing. Triggers can also be used to extend the referential integrity checks, but wherever possible, use constraints for this purpose, instead of triggers, as constraints are much faster.

Till SQL Server 7.0, triggers fire only after the data modification operation happens. So in a way, they are called post triggers. But in SQL Server 2000 you could create pre triggers also. Search SQL Server 2000 books online for INSTEAD OF triggers.

Also check out books online for 'inserted table', 'deleted table' and COLUMNS\_UPDATED()

There is a trigger defined for INSERT operations on a table, in an OLTP system. The trigger is written to instantiate a COM object and pass the newly insterted rows to it for some custom processing.**What do you think of this implementation? Can this be implemented better?**

Instantiating COM objects is a time consuming process and since you are doing it from within a trigger, it slows down the data insertion process. Same is the case with sending emails from triggers. This scenario can be better implemented by logging all the necessary data into a separate table, and have a job which periodically checks this table and does the needful.

**Sponsored Links**

<a href="http://a.collective-media.net/jump/idgt.geekinterview/article\_below;sec=article;fold=below;tile=4;sz=300x250;ord=123456789?" target="\_blank"><img src="http://a.collective-media.net/ad/idgt.geekinterview/article\_below;sec=article;fold=below;tile=4;sz=300x250;ord=123456789?" width="300" height="250" border="0" alt=""></a>

<A TARGET="\_blank" HREF="http://ad.doubleclick.net/activity;src%3D2309966%3Bmet%3D1%3Bv%3D1%3Bpid%3D64091055%3Baid%3D241454742%3Bko%3D0%3Bcid%3D41616794%3Brid%3D41634581%3Brv%3D1%3Bcs%3Df%3Beid1%3D536563%3Becn1%3D1%3Betm1%3D0%3B\_dc\_redir%3Durl%3fhttp://ad.doubleclick.net/click%3Bh%3Dv8/3b3b/3/0/%2a/h%3B241454742%3B0-0%3B0%3B64091055%3B4307-300/250%3B41616794/41634581/1%3Bu%3D%2Cidgt-10220457568\_1309833836%2C11f689b66a81fdd%2Cdatacenter%2Cax.70-am.v-am.t-am.bk-ak.b-am.h-am.e-am.a-am.b-idgt.gadgets\_m-idgt.cameras\_m-idgt.mac\_apple\_m-idgt.unicomm\_m-idgt.mobile\_phones\_l-idgt.opensource\_l-idgt.gaming\_m-idgt.cloudcomputing\_l-idgt.careers\_%3B%7Eaopt%3D0/ff/ff/ff%3B%7Efdr%3D241532105%3B0-0%3B1%3B54911015%3B4307-300/250%3B42271621/42289408/1%3Bu%3D%2Cidgt-10220457568\_1309833836%2C11f689b66a81fdd%2Cdatacenter%2Cax.70-am.v-am.t-am.bk-ak.b-am.h-am.e-am.a-am.b-idgt.gadgets\_m-idgt.cameras\_m-idgt.mac\_apple\_m-idgt.unicomm\_m-idgt.mobile\_phones\_l-idgt.opensource\_l-idgt.gaming\_m-idgt.cloudcomputing\_l-idgt.careers\_%3B%7Eaopt%3D2/1/ff/0%3B%7Esscs%3D%3fhttp://www.google.com"> <IMG SRC="http://s0.2mdn.net/2309966/PID\_1586013\_300x250\_backup.gif" width="300" height="250" BORDER="0" alt=""> </A> <IMG SRC="http://ad.doubleclick.net/activity;src=2309966;met=1;v=1;pid=64091055;aid=241454742;ko=0;cid=41616794;rid=41634581;rv=1;×tamp=6154257;eid1=9;ecn1=1;etm1=0;" width="0px" height="0px" style="visibility:hidden" BORDER="0"/> <IMG SRC="" width="0px" height="0px" style="visibility:hidden" BORDER="0"/> <IMG SRC="" width="0px" height="0px" style="visibility:hidden" BORDER="0"/>



**What is a self join? Explain it with an example.**

Self join is just like any other join, except that two instances of the same table will be joined in the query. Here is an example: Employees table which contains rows for normal employees as well as managers. So, to find out the managers of all the employees, you need a self join.

CREATE TABLE emp  
(  
empid int,  
mgrid int,  
empname char(10)  
)

INSERT emp SELECT 1,2,'Vyas'  
INSERT emp SELECT 2,3,'Mohan'  
INSERT emp SELECT 3,NULL,'Shobha'  
INSERT emp SELECT 4,2,'Shridhar'  
INSERT emp SELECT 5,2,'Sourabh'

SELECT t1.empname [Employee], t2.empname [Manager]  
FROM emp t1, emp t2  
WHERE t1.mgrid = t2.empid

Here's an advanced query using a LEFT OUTER JOIN that even returns the employees without managers (super bosses)

SELECT t1.empname [Employee], COALESCE(t2.empname, 'No manager') [Manager]  
FROM emp t1  
LEFT OUTER JOIN  
emp t2  
ON  
t1.mgrid = t2.empid

***PL/SQL***

1. IF-THEN-ELSE Statement
2. CASE Statement

select table\_name,

CASE

WHEN owner='SYS' THEN 'The owner is SYS'

WHEN owner='SYSTEM' THEN 'The owner is SYSTEM'

ELSE 'The owner is another value'

END

from all\_tables;

1. **Commit**

The syntax for the COMMIT statement is:

COMMIT [WORK] [COMMENT text];

The Commit statement commits all changes for the current session.

1. **Rollback**

The syntax for the ROLLBACK statement is:

ROLLBACK [WORK] [TO [SAVEPOINT] savepoint\_name];

The Rollback statement undoes all changes for the current session up to the savepoint specified. If no savepoint is specified, then all changes are undone.