SQL

Write a query to get last 10 records from the table.

Answer # 1

```
select * from emp
  minus
  select * from emp
   where
  rownum<=(select count(*)-10 from emp);
```

how to find out the 3 maximum salary?

```
Select * from emp where sal in (select min(sal) from (select * from

 emp order by sal desc)where rownum<=3)
```

how to find out the 2 maximum salary?

```
select max(sal)from  emp where sal not in( select max(sal)  from emp)
```

select top 3 employees salary.?

```
SELECT * FROM (SELECT * FROM emp ORDER BY sal DESC)

 WHERE ROWNUM < 4.
```

What is Difference between Having and Where clause?

Answer # 2

Having clause is used for filtering grouped data and where clause is used for filtering rows.

where clause filter the data before gruop by clause and having clause clause

filter the data after group by clause.

Retrieving Duplicate rows

```
select count(*) AS "Duplicate Rows",eid,ename,esal,edept,sid from test_emp
group by eid,ename,esal,edept,sid
having count(*) > 1
```

Filtering Duplicate rows(only original data exists)

```
select distinct eid,ename,esal,edept,sid from test_emp.
```

Query to print out the individual total number of duplicate row in sql.

Answer # 1

ANS:

```
SELECT department_id,COUNT(department_id) AS
"Occurrences"
FROM departments
GROUP BY department_id
HAVING ( COUNT(department_id) > 1 )
```

What are the difference between Functions/Stored Procs and Triggers and where are they used.

Answer # 1

Function-1.function used for calculation
perpose.it must
return value.(call by pl/sql code block,by sql
query)

Procedure-it is used to implement business
logic.it may or
not return value.it pre compile and stored in
server data
base.so it is faster.(call by pl/sql code block_

trigger-it is used to implement business logic
and maintain
intigrity.it cant take any para-meter like fun or
procedure.it is automatically fired(implicitely

How can I create a table from another table without copying any values from the old table?

Answer # 8

In oracle

```
  Create table newtablename as select * from
oldtablename where 1=2
```

Retrieve last row from table

```
select * from tablename where rowid= (select max(rowid)from tablename);
```

Retrieve First row from table

```
select * from tablename where rowid= (select min(rowid)from tablename);
```

Rename table syntax as follows:

```
alter table table_name rename to  new_table_name;
```

How to remove duplicate rows from a table?.	
Answer # 1	for example our emp table have duplicate rows than we can run this query to delete those duplicate rows Delete from emp a where rowid <> (select max(rowid) from emp where empno = a.empno);

Get the top 10 employees based on their salary

```
Select ename,esal from(Select ename,esal Rank() over(order by sal
desc)sal_rank from emp)where sal_rank<=10.
```

Select the employees making the top 10 salaries

```
SELECT ename, sal FROM ( SELECT ename, sal, DENSE_RANK() OVER
(ORDER BY sal DESC) sal_dense_rank  FROM emp )
 WHERE sal_dense_rank <= 10;
```

To get top 10 earners of the

```
select * from(select * from test_emp order by esal desc) where rownum<=10;
```

query for displaying records in a table from row 2 to row 5?

```
select * from (select rownum  num , eid, ename,esal,edept,sid,edate

from test_emp) where num between 1 and 3.
```

is it necessary to write group by and order by clause together?	
Answer # 1	No it is not necessary

Primary key:
 It is an unique identifier, it wont allow null value, in a table there will be only one column as primary key.

Composite key:
 It same as Primary key but combination of more that one column.

Write a query to find the name of employees those who have joined on Monday.(based on column hire_date)

```
SELECT FIRST_NAME,LAST_NAME FROM employees
```

```
where TRIM(TO_CHAR(TO_DATE(HIRE_DATE),'DAY'))='MONDAY';
```

Difference between IN and EXISTS?

IN - returns a numeric value.
EXIT - returns a boolean value.

Nth Highest Salary?

```
select * from test_emp where esal=( select min(esal) as Salary from

 (select distinct esal from test_emp order by esal desc) where rownum <= N)
```

Retrieving a particular row from a table?

```
SELECT * FROM test_emp WHERE rownum=1 AND rowid NOT IN

(SELECT rowid FROM test_emp WHERE rownum < N);

                (OR)

SELECT * FROM test_emp a  WHERE  N = (SELECT COUNT(rowid)

 FROM test_emp b WHERE a.rowid >= b.rowid);
```

If you want to list the tables presented in each database ?

SELECT * FROM DBA_TABLES;
or
SELECT * FROM USER_TABLES; -->
this will display all the tables of particulara db.

If you use SELECT * FROM TAB; --> this will display all the tables,views or synonyms of particulara db.

How to count the no of records of a table without using COUNT function?

```
select max(rownum) from table_name;
```

cursor is a temporary buffer used to hold the transactional
data for the manipulation purpose. data will not permanently stored in
database. it is not reusable.
syntax:
cursor <cursor name> is <select stmt);
we have two types of cursors are there.
1.implicit cursors-automatically created whenever dml
statements are executed.
2.explicit cursors-created by user

write a query to delete similar records in same table?

delete from table_name where rowid not in

(select min(rowid)from table_name group by column_name);

** column_name which having duplicate record

write a query to delete similar records in different tables with same structure?

DELETE FROM table1 WHERE(col1,col2,..) IN (SELECT col1,col2,.. FROM

table2 where condition)

write a query to delete similar records in particular fields(columns) in different tables

```
CREATE OR REPLACE  PROCEDURE DUP AS

  DECLARE

   TABLENAME_TMP  TABLE;

           CURSOR C1 IS
           SELECT M.TABLE_NAME
           FROM USER_TAB_COLS M
           WHERE M.COLUMN_NAME LIKE 'EMPNO';

           BEGIN
```

```
          OPEN C1;
            LOOP
             FETCH C1 INTO TABLENAME_TMP;
             WHEN C1%NOTFOUND THEN EXIT;

             DELETE FROM TABLENAME_TMP A WHERE ROWID NOT IN
(SELECT MAX(ROWID) FROM TABLENAME_TMP B

 WHERE A.EMPNO>=B.EMPNO);
            ENDLOOP;
          CLOSE C1;
END DUP;
```

what are the different types of joins?

1)Self Join
2)Equi join (or)Inner Join
       2.1)Non-equi Join
3)Outer Join
  3.1) Left-Outer Join
  3.2) Right-Outer Join
  3.3) Full-Outer Join

what command is used to create a table by copying the structure of another table including constraints ?

We can only copy the structure of the table and not the constraints. In this way only NOT NULL constraint specified on any column of the OLDTABLE will automatically get copied into the NEWTABLE but other constraint will have to be created manually.

create table newtablename as select *from oldtablename;

with the above query total data will also be copied.

But to copy only the structure.

create table newtablename as select * from oldtablename where 1=2;

(table structure only copied)

When we can declare a column as Unique and Not Null both at the same time. What is the use pf Primary Key then?

The concept of adding an unique key and not null constraint to a column as

well as adding as unique index to it , you will assume to get the benefits of adding an primary key to that column.
But use it for the purpose of referential integrity will not allow to do so.
That's what PK is so useful. You will use it for normalization purpose.

Unique Key creates Non-Cluster index in default,
But in Primary Key creates Cluster index in default

How do you get all records from 2 tables. Which join do you use?

Using FULLOUTER JOIN?

triggers are invoked implicitly , while stored procedures are invoked explicitly.

What is the fastest way of accessing a row in a table?	
Answer # 1	The fastest way to access row is by rowid.

I have one table :EID,Depid,ename,sal I want to have max(sal) in each department.?

select depid,max(sal) from <table name> group by depid;

write a query to dispaly those name who is more than one in student table? exampple- in a student table sandeep kumar comes 4 times, rakesh kumar comes 2 times, ajit kumar comes 1 times so query will display sandeep kumar and rakesh kumar single times?	
Answer # 1	SELECT COUNT(NAME_FIELD_NAME)AS NOMBER_OF_OCCURENCE FROM STUDENT WHERE NUMBER_OF_OCCURANCE >1;

display list of all users from ur data base.what is the query?

select * from dba_users

select * from user_tables;

i deleted 4 records form the table.after i have applied commit .now i want to get back those records.how?

Flashback table feature is available from Oracle 9i onwards only. Before that deleted records once committed cannot be rolled back. As far as Oracle 9i onwards the solution specified by Bipin Maurya is correct. But i would like to add something. For using FLASHBACK TABLE the user must be granted privilege for the same. But there is another way out.

```
create table temp_new as select * from temp as of timestamp
to_timestamp('11/01/2011 08:45:00','dd/mm/yyyy hh24:mi:ss');
```

This will bring the table to the point in time specified by timestamp.

Using flashback table:

```
# enable row movement
1) alter table test2 enable row movement;
# flashback test2 to 10 minutes ago.
2) FLASHBACK TABLE test2 TO TIMESTAMP SYSTIMESTAMP -
INTERVAL '10' MINUTE;
```

how many outer joins are used in a single sql query? is there any limitations for using of outer joins?

Answer # 1	ANS: Only one outer join is possible for a single query & it can be either left outer join or right outer join or full outer join. We can't use two outer joins for a single query.

Flash Table

Use the FLASHBACK TABLE statement to restore an earlier state of a table in the event of human or application error. you must have either the FLASHBACK object privilege on the table or the FLASHBACK ANY TABLE system privilege. In addition, you must have the SELECT, INSERT, DELETE, and ALTER object privileges on the table. Row movement must be enabled for all tables in the Flashback list.

Create table employees_demo, with row movement enabled, from table employees of the sample hr schema:

```
CREATE TABLE employees_demo ENABLE ROW MOVEMENT AS SELECT *
FROM employees;
```

Restore the table employees_demo to its state 1 minute prior to the current system time:

FLASHBACK TABLE employees_demo TO TIMESTAMP (SYSTIMESTAMP - INTERVAL '1' minute);

Retrieving a Dropped Table: Example

If you accidentally drop the hr.employees table and wish to retrieve it, issue the following statement:
FLASHBACK TABLE employees TO BEFORE DROP;

If another employees table has been created in the hr schema, use the RENAME TO clause to rename the retrieved table:
FLASHBACK TABLE employees TO BEFORE DROP RENAME TO employees_old;

If you know that the employees table has been dropped multiple times, and you wish to retrieve the oldest version, query the USER_RECYLEBIN table to determine the system-generated name, and then use that name in the FLASHBACK TABLE statement. (System-generated names in your database will differ from those shown here.)
SELECT object_name, droptime FROM user_recyclebin WHERE

original_name = 'employees';

Display full details for the creditor/s who has received the single largest payment. Do not use a table join or set operator anywhere in your query.

Answer # 4	SELECT MAX(SALARY) FROM (SELECT FIRST_NAME,SALARY,FROM EMP ORDER BY SALARY DESC) WHERE ROWNUM<=1 GROUP BY FIRST_NAME,SALARY;

Diference between Oracle 9i and 10G.

Answer # 1	Major changes to SQL optimizer internals Oracle Grid computing AWR and ASH tables incorporated into Oracle Performance Pack and Diagnostic Pack options Automated Session History (ASH) materializes the Oracle Wait Interface over time

Data Pump replaces imp utility with impdp

Automatic Database Diagnostic Monitor (ADDM)

SQLTuning Advisor

SQLAccess Advisor

Rolling database upgrades (using Oracle10g RAC)

dbms_scheduler package replaces dbms_job for scheduling

Top 5 salaried employees in each dept?

```
select * from emp X where 5> (select count(*) from emp

where edept = X.edept and esal > X.esal) order by edept,esal desc
```

In oracle 10g :

```
SELECT * FROM (SELECT EID,EDEPT,ENAME, ESAL, DENSE_RANK()

OVER(PARTITION BY EDEPT ORDER BY ESAL DESC) RANK FROM EMP);
```

HOW TO RETRIVE FIRST AND LAST RECORDS FROM A TABLE?	

Answer # 1	SELECT * FROM EMP WHERE ROWID IN(SELECT MIN(ROWID) FROM EMP)
	UNION ALL
	SELECT * FROM EMP WHERE ROWID IN(SELECT MAX(ROWID) FROM EMP);

(OR)

```
select * from employees where rowid = (select max(rowid) from employees)or
rowid =(select min(rowid)from employees)
```

Queery for retrieving required records?

```
SELECT column FROM table WHERE ROWNUM <= N
```

(OR)

```
 select * from (select rownum r,emp.* from emp)

where  r between 1 and N;
```

how to delete all duplicate records from a table using subquery?	

Answer # 4	DELETE FROM TableName
	WHERE ROWID IN (SELECT ROWID FROM
	TableName WHERE ROWID NOT IN(SELECT
	MAX(ROWID) FROM
	TableName GROUP BY Column1, Column2));

What are main difference between Stored Procedure and Functions.?

stored procedure : It is Pre Compiled Code(PCode) it is stored in Database. we can reuse without compile.

Function: It need compile whenever it will run.
Functions should have RETURN clause..

functions:
1.The parameters are input values and output values
2.The functions will return a value
3.The functions will be called with in sql

Procedures:
1.The parameters are input values and output values
2.The procedures will not return any value
3.The procedures will not be called with in sql

Query for last N Records?

SELECT * FROM emp e WHERE N >(SELECT COUNT(1) FROM emp f WHERE e.rowid<f.rowid);

There are two tables (T1, T2) with different rows. Find the output of query. Select count(*) from T1,T2.	
Answer # 2	yes multiply both tables rows

how to find Nth row ?(N th Record)

select * from emp where rownum<=N minus select * from emp where

rownum<N

(or)

```
select * from emp e1 where N = (select count(rowid)from emp e2

where e1.rowid >= e2.rowid);
```

View is nothing but the virtual table made from few columns of the base table to get faster retrieval and fetch of the data. Hence when you update any view the column which you updating is get updated, and the column basically belongs to the base table. hence answer is IT WILL UPDATE BASE TABLE.

Display database name?

```
select name from v$database.
```

Nth MAX Sal in a one query?

```
select distinct(a.sal) from emp a where &n=(select count(distinct(b.sal)) from
emp b where  a.sal<=b.sal);
```

(OR)

for max

```
select a.sal from emp a where 1=(select distinct(count(b.sal))from

emp b where a.sal<=b.sal);
```

for second max

```
select a.sal from emp a where 2=(select distinct(count(b.sal))from emp

 b where a.sal<=b.sal)
```

For Nth Max.

```
select distinct a.sal from emp a where &N= (select count(distinct b.sal) from

emp b where b.sal>=a.sal)
```

For Nth Min.

```
select distinct a.sal from emp a where &N= (select count(distinct b.sal) from
emp b where b.sal<=a.sal)
```

query for top max salaries?

select * from (select * from emp order by esal desc) where rownum<=N;

primary key is used to identify a row and it doesn't allow null values. It avoids duplication of rows. Primary key will not allow "Null values" and "Duplicate values". Primary key is unique,primary key is not NULL .

foreign key is NULL,foreign key reference as Primary key in another table. whereas foreign key refers to a column or combination of columns included in the defenition of referential integrity.
Foreign key will allow "Null values" and "Duplicte values"and it refers to a primary key in anoter table.

what are pseudocolumns?	
Answer # 4	Pseudocolumns are not actual columns in a table but they behave like columns. For example, you can select values from a pseudocolumn. However, you cannot insert into, update, or delete from a pseudocolumn. Also, pseudocolumns are allowed in SQL statements, but not in procedural statements. PL/SQL recognizes the following SQL pseudocolumns, which return specific data items: CURRVAL, LEVEL, NEXTVAL, ROWID, and ROWNUM.

Re: What is the difference between column level constraints and table level constraints?	
Answer # 2	column level constraints contain all types of constraints (like, not null,primary key,foregin key,unique).but table level except not null constraint its supports all constraints.

What is difference between UNIQUE constraint and PRIMARY KEY constraint ?

Answer # 1

The Primary key Constraint= unique + Not Null constraints.
For the Unique the meaning it self it is telling unique value. won't accept duplicate values.

What are Clusters ?

Answer # 1

Cluster is amechanism which is used to bind data together such that it will improve performance while retreiving information from a table

Triggers are the database objects (Like Procedures) which will be executed by the Database itself and cannot be excuted by the user explicitly. The trigger can be written to execute while inserting/ updating/ deleting a row.

Two types of triiger

ROW LEVEL TRIGGER
===============
Row Level triggers are FOR EACH ROW, that is, the trigger is activated for each row that is inserted/updated/deleted.

STATEMENT LEVEL TRIGGER
=====================
Statement level triggers will execute the action specified only once. (Will not consider the no. of rows being inserted/updated/deleted)

Difference between IN and EXISTS

Answer # 3

IN - returns a numeric value.
EXIT - returns a boolean value

can we call a procedure from a function?

Yes i am aggree with all three answers but there are diffrent Scenarios.you can not call a procedure in which dml ,ddl statement or transaction controling statement is used.you can all a procedure in which only select statement is used.please chekc if you dont have trust.

Re: There is a table1 with records (1,2,3,4,5,6) and table2 with records (4,5,6,7,8,9).write a query so as to get the result as 1,2,3,4,5,6,7,8,9

Answer # 1

select * from table1 union select * from table2

-- Union will give distinct value of both tables

Can we delete the trigger in a view? If yes why if not why?

Answer # 2

As per my knowledge we can drop trigger crated on view see below exp

```
CREATE OR REPLACE FORCE VIEW
"SCOTT"."TEST_VIEW"
("EMPNO", "ENAME", "JOB", "MGR",
"HIREDATE", "SAL", "COMM",
"DEPTNO") AS
select  "EMPNO", "ENAME", "JOB", "MGR",
"HIREDATE", "SAL", "
COMM", "DEPTNO" from emp;

SQL> create or replace
  2  TRIGGER SCOTT.VIEW_TIG
  3  INSTEAD OF INSERT OR DELETE OR
UPDATE ON TEST_VIEW
  4  BEGIN
  5    NULL;
  6  END;
  7  /

Trigger created.

SQL> drop trigger VIEW_TIG ;

Trigger dropped.
```

What are the factors you will check for the performane optimization for a database query?

1)The most important option to increase the query performance is to create index on the table.
First one should concerntrate on the columns used in the where cluase.

But indexes can hit the performance for insert queries. So before giving an insert query, better to remove the index and then to recreate the index.

2)

1.In the select statement give which ever the columns you need

don't give select * from
2.Do not use count()function
3.use indexes and drop indexes regularly it B tree structure we are not going to drop means it will take time for creating indexes it self.
4.when joining the tables use the id's(integer type)not string
5.Sometimes you may have more than one subqueries in your main query. Try to minimize the number of subquery block in your query.
6.Use operator EXISTS, IN and table joins appropriately in your query.
a) Usually IN has the slowest performance.
b) IN is efficient when most of the filter criteria is in the sub-query.
c) EXISTS is efficient when most of the filter criteria is in the main query.
7.Use EXISTS instead of DISTINCT when using joins which involves tables having one-to-many relationship.
8.Try to use UNION ALL in place of UNION.
9.Avoid the 'like' and notlike in where clause.
10.Use non-column expression on one side of the query because it will be processed earlier.
11.To store large binary objects, first place them in the file system and add the file path in the database.

3)
try to use function based indexes and use it when needed..
it will not be used unnecessorily.
use clustering concept while creating table it will make you join operation fast if more join operation u r going to
perform on the tables

What is data abstraction?

The system hides certain details of how data is stored and created and maintained Complexity should be hidden from database users.this is data abstraction.

How will you fine tune a stored procedure or what are the steps that should be taken to fine tune or optimize a stored procedure?

# Use stored procedures instead of heavy-duty queries.
This can reduce network traffic, because your client will
send to server only stored procedure name (perhaps with some
parameters) instead of large heavy-duty queries text. Stored
procedures can be used to enhance security and conceal
underlying data objects also. For example, you can give the
users permission to execute the stored procedure to work
with the restricted set of the columns and data.

# Include the SET NOCOUNT ON statement into your stored
procedures to stop the message indicating the number of rows
affected by a Transact-SQL statement.
This can reduce network traffic, because your client will

not receive the message indicating the number of rows affected by a Transact-SQL statement.

# Call stored procedure using its fully qualified name.

The complete name of an object consists of four identifiers: the server name, database name, owner name, and object name. An object name that specifies all four parts is known as a fully qualified name. Using fully qualified names eliminates any confusion about which stored procedure you want to run and can boost performance because SQL Server has a better chance to reuse the stored procedures execution plans if they were executed using fully qualified names.

# Consider returning the integer value as an RETURN statement instead of an integer value as part of a recordset.

The RETURN statement exits unconditionally from a stored procedure, so the statements following RETURN are not executed. Though the RETURN statement is generally used for error checking, you can use this statement to return an integer value for any other reason. Using RETURN statement can boost performance because SQL Server will not create a recordset.

# Don't use the prefix "sp_" in the stored procedure name if you need to create a stored procedure to run in a database other than the master database.

The prefix "sp_" is used in the system stored procedures names. Microsoft does not recommend to use the prefix "sp_" in the user-created stored procedure name, because SQL Server always looks for a stored procedure beginning with "sp_" in the following order: the master database, the stored procedure based on the fully qualified name provided, the stored procedure using dbo as the owner, if one is not specified. So, when you have the stored procedure with the prefix "sp_" in the database other than master, the master database is always checked first, and if the user-created stored procedure has the same name as a system stored procedure, the user-created stored procedure will never be executed.

# Use the sp_executesql stored procedure instead of the EXECUTE statement.

The sp_executesql stored procedure supports parameters. So, using the sp_executesql stored procedure instead of the EXECUTE statement improve readability of your code when there are many parameters are used. When you use the sp_executesql stored procedure to executes a Transact-SQL statements that will be reused many times, the SQL Server query optimizer will reuse the execution plan it generates for the first execution when the change in parameter values to the statement is the only variation.

# Use sp_executesql stored procedure instead of temporary stored procedures.
Microsoft recommends to use the temporary stored procedures when connecting to earlier versions of SQL Server that do not support the reuse of execution plans. Applications connecting to SQL Server 7.0 or SQL Server 2000 should use the sp_executesql system stored procedure instead of temporary stored procedures to have a better chance to reuse the execution plans.

# If you have a very large stored procedure, try to break down this stored procedure into several sub-procedures, and call them from a controlling stored procedure.
The stored procedure will be recompiled when any structural changes were made to a table or view referenced by the stored procedure (for example, ALTER TABLE statement), or when a large number of INSERTS, UPDATES or DELETES are made to a table referenced by a stored procedure. So, if you break down a very large stored procedure into several sub-procedures, you get chance that only a single sub-procedure will be recompiled, but other sub-procedures will not.

# Try to avoid using temporary tables inside your stored procedure.
Using temporary tables inside stored procedure reduces the chance to reuse the execution plan.
# Try to avoid using DDL (Data Definition Language) statements inside your stored procedure.
Using DDL statements inside stored procedure reduces the chance to reuse the execution plan.
# Add the WITH RECOMPILE option to the CREATE PROCEDURE statement if you know that your query will vary each time it is run from the stored procedure.
The WITH RECOMPILE option prevents reusing the stored procedure execution plan, so SQL Server does not cache a plan for this procedure and the procedure is recompiled at run time. Using the WITH RECOMPILE option can boost performance if your query will vary each time it is run from the stored procedure because in this case the wrong execution plan will not be used.

# Use SQL Server Profiler to determine which stored procedures has been recompiled too often.
To check the stored procedure has been recompiled, run SQL Server Profiler and choose to trace the event in the "Stored

Procedures" category called "SP:Recompile". You can also trace the event "SP:StmtStarting" to see at what point in the procedure it is being recompiled. When you identify these stored procedures, you can take some correction actions to reduce or eliminate the excessive recompilations.

Re: what is sql injection in sql server?

Answer # 1	SQL injection is something where an end user of an application is able to submit SQL queries through the user interface and end up running those queries, causing potential damage to the database. Is This Answer Correct ? 0 Yes 0 No	0	Pankaj Kumar

Re: what is sql injection in sql server?

Answer # 2

SQL injection is nothing but inserting malicious code with
the strings and later which will pass to the particular SQL
server instances which may damage the database or may
corrupt bthe data

For exaple:

```
var EmpName;
EmpName= Request.form ("EmpName");
var sql = "select * from Employee where
EmpName= '" +
EmpName+ "'";
```

If you considers the above query and if user prompts to
enter the input as Scott,

Then query processed like this

```
select * from Employee where
EmpName='Scott'
```

If user enters the input as below,

Scott,drop table Employee
In this case your query will be processed
as below

select * from Employee where
Ename='Scott',drop table emp

So,first select statement will be executed
and then table
will be dropped.

Nothing but without standard of coding an
expertised user
may damages or corrupt the databases.

To avoid SQL injection attacks:-
1)Use Parameterized Input with Stored
Procedures
2)Use the Parameters Collection with
Dynamic SQL
3)Filtering Input
4)LIKE Clauses

What is the difference Delete and Drop ?

Answer # 3

1.DELETE COMMANDS: ONLY DELETE DATA
FROM TABLE NOT THE
STRUCTURE OF THE TABLE.DATA CAN BE
CALL BACK USING ROLL
BACK.

2.DROP COMMANDS DLETE TABLE AND
STRUCTURE BOTH FROM THE
DATABASE ie.DATA CANT BE ROLL BACK
OVER HERE.

cursor is temperory block of memory which holds the data as temperory in buffer. Cursor is an named private area called active set which is used to process the multiple rows returned by the query.cursors are mainly used in procedure/package/function when
the select qurey in procedure/package/function returns more than one value.

Let?s say you have 100,000 records and you want to delete 95,000 at a time and keep only 5 thousand. But in local memory you don?t have

enough space for 95,000 records. What do you do in this case? How do you delete without any problem with time and performance?	
Answer # 1	First select the 5000 records u want to keep.. put it in memory.. then delete the entire table.. and then put this 5000 record table in place

Can we create index on views?

An index cannot be defined on a view. Because view is virtual table, which consists of a subset of columns from one more tables.

The Restrictions imposed on views are as follows:
1. A view can be created only in the current database.
2. A view can be created only if there is a SELECT permission on its base table
3. A Trigger or an Index cannot be defined on a view.
4. A view cannot be derive its data from temporary tables.
5. The CREATE VIEW statement cannot be combined with other SQL statements in a single batch.

What is the difference between implicit index and explicit index?	
Answer # 1	Implicite index is created by oracle Internally ex: when we make a primary key in a Column then implicit Index created which is Clustered Type index(Default) while Explicite index created by "Create Index Command' User

can foreign key take role of primary key?

Yes. This situation occurs in one-to-one relationships.

For example:

```
create table Table1
( T1_Id    integer     not null primary key
, T1_Data  varchar(9)  not null
)
create table Table2
( T2_Id    integer     not null primary key
, T2_Data  varchar(37) not null
, foreign key (T2_Id) references Table1 (T1_Id)
)
```

Why would someone design two tables like this? If it's a one-to-one relationship, couldn't you just include T2_Data in Table1? Yes, you could, although then you would also need to allow T2_Data to be NULL, if not every T1_Id has a matching T2_Id—in this case it's actually a one-to-zero-or-one relationship.

If it's a true one-to-one relationship, where every T1_Id has a matching T2_Id, then you could combine them without NULLs. Then the only reason why you might want separate tables would be if T2_Data were (a) infrequently queried, and (b) so large that it would result in many fewer physical rows on a physical database page, which would imply poorer performance for queries that used only T1_Data.

How can you find the size of a database?	
Answer # 2	select sum(bytes/1024/1024) from dba_data_file; it is the size of totaldatafiles

diffrence between Co-related sub query and Nested Subquery?

Correlated subquery runs once for each row selected by the outer query. It contains a reference to a value from the row selected by the outer query.
Nested subquery runs only once for the entire nesting (outer) query. It does not contain any reference to the outer query row.

Correlated Subquery:

select e1.empname e1.basicsal e1.deptno from emp e1 where e1.basicsal = (select max(basicsal) from emp e2 where e2.deptno = e1.deptno)

Nested Subquery:

select empname basicsal deptno from emp where (deptno, basicsal) in (select deptno max(basicsal) from emp group by deptno)

different between Stored Procedure and Procedure?
In Layman tounge -

Whenever any named block in PL/SQL is created using CREATE OR REPLACE clause, and as we compile it will get stored in database, henceforth other code snippets can call/use this block which ulimately resides in database after creation i.e. "stored in database". And thus will move along database if ever we move the database.

In oracle way -

A stored procedure is created and stored in the database as a schema object. Once created and compiled, it is a named object that can be run without recompiling. Additionally, dependency information is stored in the data dictionary to
guarantee the validity of each stored procedure.
Once the procedure is created and compiled ,it
is stored in database permanently.That's why it is called as stored procedure.
How to find 1st, 2nd, 3rd highest using select select statement not using rownum?

```
SELECT sal FROM(SELECT sal,RANK()OVER(ORDER BY sal DESC) AS rnk
FROM  emp)WHERE rnk IN (1,2,3);
```

difference between first normal form & second normal form?

First normal form (1NF or Minimal Form) is a normal form used in database normalization. A relational database table that adheres to 1NF is one that is free of repeating groups.Second normal form (2NF) is a normal form used in database normalization,where a table that is in first
normal form (1NF) must meet additional criteria if it is to qualify for second normal form. Specifically:a 1NF table is in 2NF if and only if none of its non-prime attributes are functionally dependent on a part (proper subset) of a candidate key. (A non-prime attribute is one that does not
belong to any candidate key.)

First normal form, sets the very basic rules for an
organized database i.e. Eliminate duplicative columns. Create separate tables for each group of related data and identify the row with a unique column i.e the primary key.

Second normal form further removes the duplicate data: i.e. Meet the requirements of the first form of normalalization. Create relationships between new tables through foreign keys.

difference between truncate and delete ,drop?

truncate

it deletes only the records from the table
it doesn't delete the structure of the table
we cannot rollback it
we cannot delete a particular record from the table

it is ddl

drop

it deletes the records as well as structure of the table too
we cannot rollback the records
we can flashback the table from the version 9x
it is ddl

delete

it deletes the only records which are we mention in the where clause or whole records
it is dml
we can rollback it

Can Having clause be used without Group by clause?

HAVING Clause Specifies a search condition for a group or an aggregate. HAVING is usually used with the GROUP BY clause. When GROUP BY is not used, HAVING behaves like a WHERE clause.

What is the order in which the SQL query is executed? list them in order.?

When a SQL query is getting executed it goes through the following steps:

1) It goes to the shared pool which contains information such as the parse tree and execution plan for the corresponding statement.
2)Validate the SQL statement.
3)Validate the source.
4)Acquire locks.
5)Checks for previllages.
6)Parse the statement.
7)Execute the Query.
8)Fetch the value (only for select statement).
9)Generate the redo information

(2)

At runtime the compiler first takes the table name from where the data is to be retrieved (FROM <tablename>), then it checks for the condition (Where <condition>) then it selects the values to me displayed as o/p

What is SQL Profiler and the use of it?

To check how much time a query is going to execute.

What is the difference between UNION and UNIONALL?	
Answer # 2	union is used to retreive the data with out duplication union all is used to retreive the data with duplication

What are SQL Jobs, please mention the use of it and how to create them.

sql job is scheduling the work into server and it will automatically run and finish the job for this we have to set the time.what is the use of the job is
1) it check the database automatically
2)take a back up of database
for this job we have to start the Sqlserver agent and give the date and time for this particular job

IF i write before / after insert / update trigger and i do rollback what will happen?

A trigger May Not issue A TCL statement like COMMIT,ROLLBACK,SAVEPOINT.So U Cant Write A TCL Command Inside A trigger.

How to get the count of distinct records. Please give me the query?	
Answer # 6	select count(distinct(columnname)) from tablename

What are explicit and implicit transactions?

When a connection is operating in implicit transaction mode MS SQL automatically starts a new transaction after the current transaction is committed or rolled back.Implicit transaction mode generates a continuous chain of transactions.

An explicit transaction is one in which you explicitly
define both the start and end of the transaction.A user has to use the BEGIN TRANSACTION, COMMIT TRANSACTION, COMMIT WORK, ROLLBACK TRANSACTION, or ROLLBACK WORK Transact-SQL statements to define explicit transactions.

display 1 to 10 numbers using one select statement?

select level from dual connect by level<=10.

what is Normalization?	
Answer # 6	Normalizatiin is the scientific method of breaking down complex table sturcture into simple table structure by using certain rules.using this method you can reduce redundency in a table and eliminate the problem of inconsisteny and disk space usage

if UNION ALL and Natural Join does the same operation and are same?

Both are not same why because in UNION ALL operator can retrive the total number of records from the table and NATUAL JOIN OR EQUI JOIN operator retrive the rows which are condition must be the same and datatype of the column names must be same

emp table

```
empid ename salary
1   a  1000
2   b  2000
```

dept table

```
deptid deptno
1 10
2 20
```

if i join this two tables
 (empid = deptid)

```
empid ename salary deptno
1     a     1000   10
2     b     2000   20
```

if i union a11 this two tables

```
empid ename salary deptno
1     a     1000   null
2     b     2000   null
1    null   null    10
2    null   null    20
```

so join and unionall is not same

how to write the query to select the rows are in the order of either 1,3,5,7... or 2,4,6,8,.?

For 1,3,5,7...

```
select * from emp where (rowid,1) in (select rowid,mod(rownum,2) from emp);
```

For 2,4,6,8,...

```
select * from emp where (rowid,0) in (select rowid,mod(rownum,2) from emp);
```

If a table column has is UNIQUE and has NOT NULL, is it equivalent to a PRIMARY KEY column?

No. It will same for doesn't allow null and duplicate values. But, Primary key constrained applied Cluster index on that column and unique key constraint non clustered index on that column. that is the major differece still be there.

Remove duplicate Rows (modified)

```
delete from table_name where rowid not in(select min(rowid)from table_name
groupcolumn1,column2,column3)
```

What meant by Performance Tuning,how can we do the performance tuning on stored procedures and tell some steps to do the performance tuning?

Shortly : The performance tuning is reduce the load of the database.

stored procedure:

```
--create a sample performance tuning procedure

CREATE PROCEDURE dbo.spTestDelegator (@query bit) AS
BEGIN

--NOCOUNT is use to reduce the load
--if its off the procedure get (full memory) of execution

SET NOCOUNT ON;
IF @query = 0
EXEC spTestFromAuthors
ELSE
EXEC spTestFromPublishers
GO
CREATE PROCEDURE dbo.spTestFromAuthors AS
SELECT * FROM authors
GO
CREATE PROCEDURE dbo.spTestFromPublishers AS
SELECT * FROM publishers
GO
```

STEPS:

1.Good code ideas
2.Reduces use more query in same procedure
3.use clear structure of code
4.always done stored procedure is good. because,Transact-Sql get load of execution

What is cursor ? And what is difference between Trigger ?

Cursor is an Database object and retrive the rows from database row by row ,And it is maily use for ot reduce the network traffic.

it contains 5 features

1.DECLARE the Cursor
2.OPEN the cusrsor
3.FETCH the cursor
4.CLOSE the cursor
5.DEALLOCATE the cursor

TRIGGER : IT is also an database object and You can perform an anction Trigger will fire automatacallly.like (DML,DDL)

They can cascade changes through related tables in the database; however, these changes can be executed more efficiently using cascading referential integrity constraints.

They can guard against malicious or incorrect INSERT, UPDATE, and DELETE operations and enforce other restrictions that are more complex than those defined with CHECK constraints.

Multiple DML triggers of the same type (INSERT, UPDATE, or DELETE) on a table allow multiple, different actions to take place in response to the same modification statement.

What is integrity constraints?

Data integrity allows to define certain data quality requirements that the data in the database needs to meet. If a user tries to insert data that doesn't meet these requirements, Oracle will not allow so.

Constraint types

There are five integrity constraints in Oracle.

Not Null

A column in a table can be specified not null. It's notpossible to insert a null in such a column. The default is null. So, in the following create table statement, a null can be inserted into the column named c.

```
create table ri_not_null
 (
  a number not null,
  b number     null,
  c number
  );

insert into ri_not_null values (   1, null, null);
insert into ri_not_null values (   2,    3,    4);
insert into ri_not_null values (null,    5,    6);
```

The first to records can be inserted, the third cannot,throwing a ORA-01400: cannot insert NULL into("RENE"."RI_NOT_NULL"."A").
The not null/null constraint can be altered with

```
alter table ri_not_null modify a null;
```

After this modification, the column a can contain null values.

Unique Key

The unique constraint doesn't allow duplicate values in a column. If the unique constraint encompasses two or more columns, no two equal combinations are allowed.

```
create table ri_unique
 (
  a number unique,
  b number
 );
```

However, if a column is not explicitely defined as not null, nulls can be inserted multiple times:

```
insert into ri_unique values (4,   5);
insert into ri_unique values (2,   1);
insert into ri_unique values (9,   8);
insert into ri_unique values (6,   9);
insert into ri_unique values (null,9);
insert into ri_unique values (null,9);
```

Now: trying to insert the number 2 again into a:

```
insert into ri_unique values (2,7);
```

This statement issues a ORA-00001: unique constraint(RENE.SYS_C001463 violated). Every constraint, by the way,
has a name. In this case, the name is: RENE.SYS_C001463.
In order to remove that constraint, an alter table ... drop constraint ... is needed:

```
alter table ri_unique drop constraint sys_c001463;
```

Of course, it is also possible to add a unique constraint on an existing table:

```
alter table ri_unique add constraint uq_ri_b unique (b);
```

A unique constraint can be extended over multiple columns:

```
create table ri_3
 (
  a number,
  b number,
  c number,
  unique (a,b)
);
```

It is possible to name the constraint. The following example creates a unique constraint on the columns a and b and names the constraint uq_ri_3.

```
create table ri_3
 (
  a number,
  b number,
  c number,
  constraint uq_ri_3 unique (a,b)
);
```

Primary Key

On a technical level, a primary key combines a unique and a not null constraint. Additionally, a table can have at most one primary key. After creating a primary key, it can be referenced by a foreign key.

```
create table ri_primary_key
 (
  a number primary key,
  b number
 );
```

Primary keys can explicitely be named. The following create table statement creates a table with a primary key whose name is pk_name.

```
create table ri_primary_key_1
(
  a number,
  b number,
  c number,
  constraint pk_name primary key (a, b)
);
```

Foreign Key

A foreign key constraint (also called referential integrity constraint) on a column ensures that the value in that column is found in the primary key of another table.If a table has a foreign key that references a table, that referenced table can be dropped with a drop table cascade constraints.It is not possible to establish a foreign key on a global temporary table. If tried, Oracle issues a ORA-14455:
attempt to create referential integrity constraint on temporary table.Check A check constraint allows to state a minimum requirement for the value in a column. If more complicated requirements are

desired, an insert trigger must be used.
The following table allows only numbers that are between 0 and 100 in the column a;

```
create table ri_check_1 (
  a number check (a between 0 and 100),
  b number
);
```

Check constraints can be added after a table had been created:

```
alter table ri_check_1
  add constraint ch_b check (b > 50);
```

It is also possible to state a check constraint that check the value of more than one column. The following example makes sure that the value of begin_ is smaller than the value of end_.

```
create table ri_check_2 (
  begin_   number,
  end_     number,
  value_   number,
  check (begin_ < end_)
);
```

How to select Distinct columns from the table?

```
select distinct * from tablename;
```

IN - used to select multiple rows based on any of the key provided

SQL - select distinct employeeid from orders where orderid in (select orderid from orderdetails where discount >= 10)

ANY - used in case of relational queries to compare result with any of the key.

SQL - select custID from orders where regionID != "E" and discount > any (select discount from orders where regionID = "E" and discount > 3)

ALL - used in case of relational queries to compare result with all of the keys.

SQL - select custID from orders where regionID != "E" and discount > all (select discount from orders where regionID = "E" and discount > 3)

IN- It will return the value based on the parameter;
select * from emp where salary in ('1000','3000');

ANY-It will compare with any value that has been returned by the parameter;

select * from emp where salary > any(select salary from emp where deptno=10)
the salary will be compared with any value that has been returned by the subquery.

ALL-It will compare with max/min value that has been returned by the subquery;

select * from emp where salary > all(select salary from emp where deptno=10)
the salary will be compared with the longest value that has been returned by the subquery.

WHAT IS INDEXING?

A database index is a data structure that improves the speed of operations in a table. Indices can be created using one or more columns, providing the basis for both rapid random lookups and efficient ordering of access to records.

how to find Nth lowest salary?

SELECT DISTINCT (A.esal) FROM EMP A WHERE &N = (SELECT COUNT (DISTINCT (b.esal)) FROM EMP B WHERE a.esal>=b.esal);

what is the order of execution of where,having,group by in select stement?

The order of syntax for SELECT statement is

Select Top| Distinct From Join
On Where Group By Having Oder By

But the order of execution of SELECT statement is
1.From
2.On
3.Join
4.Where
5.Group By
6.Having
7.Distinct
8.Select
9.Order By
10.Top

how many joins we can write if at all we have n no of tables?

Answer # 1	N-1

I have one Excel file with 80,000 Records. Now I need to load that whole file into Oracle Database with same columns in Excel sheet . How can i do that?

An excel file cannot store more than 64K records.
To load an excel file, we will have to convert it to a
delimited format. the load it using sqlldr.

difference between VARCHAR, VARCHAR2 and CHAR data types?

CHAR should be used for storing **fix length character strings. VARCHAR behaves exactly the same as VARCHAR2**. However, this type should not be used as it is reserved for future usage.
VARCHAR2 is used to store **variable length character strings**. The string value's length will be stored on disk with the value itself. VARCHAR can store up to 2000 bytes of characters while VARCHAR2 can store up to 4000 bytes of characters.
If we declare datatype as VARCHAR then it will occupy space for NULL values, In case of VARCHAR2 datatype it will not occupy any space.

Can one retrieve only rows X to Y from a table?

```
select * from(select row_number() over(order by 1) rm,emp.* from emp)

where rm between x and y-------x,y are rows no.
```

Difference b/w 2 dates?

```
CREATE TABLE dates (date1 DATE, date2 DATE);

INSERT INTO dates VALUES (SYSDATE, SYSDATE-1);

INSERT INTO dates VALUES (SYSDATE, SYSDATE-1/24);

INSERT INTO dates VALUES (SYSDATE, SYSDATE-1/60/24);
```

(1)

```
  SELECT floor((date1-date2)*24)
       || ' HOURS ' ||
       mod(floor((date1-date2)*24*60),60)
```

```
        || ' MINUTES ' ||
        mod(floor((date1-date2)*24*60*60),60)
        || ' SECS ' time_difference
   FROM dates;
```

(2)

```
SELECT to_number( to_char(to_date('1','J') +
(date1 - date2), 'J') - 1)  days,
to_char(to_date('00:00:00','HH24:MI:SS') +
(date1 - date2), 'HH24:MI:SS') time FROM dates;
```

(3)

```
select numtodsinterval(date1-date2,'day') time_difference from dates;
```

1. What is a CO-RELATED SUBQUERY

A CO-RELATED SUBQUERY is one that has a correlation name as table or view designator in the FROM clause of the outer query and the same correlation name as a qualifier of a search condition in the WHERE clause of the subquery. eg

```
   SELECT  field1 from table1 X
   WHERE  field2>(select avg(field2) from table1 Y
                              where
                              field1=X.field1);
```

(The subquery in a correlated subquery is revaluated for every row of the table or view named in the outer query.)

2. What are various joins used while writing SUBQUERIES

Self join-Its a join foreign key of a table references the same table.

Outer Join--Its a join condition used where One can query all the rows of one of the tables in the join condition even though they don't satisfy the join condition.

Equi-join--Its a join condition that retrieves rows from one or more tables in which one or more columns in one table are equal to one or more columns in the second table.

3. What are various constraints used in SQL

NULL NOT NULL CHECK DEFAULT

4. What are different Oracle database objects

TABLES VIEWS INDEXES SYNONYMS SEQUENCES TABLESPACES etc

5. What is difference between Rename and Alias

Rename is a permanent name given to a table or column whereas Alias is a temporary name given to a table or column which do not exist once the SQL statement is executed.

6. What is a view

A view is stored procedure based on one or more tables, its a virtual table.

7. What are various privileges that a user can grant to another user

SELECT CONNECT RESOURCE

8. What is difference between UNIQUE and PRIMARY KEY constraints

A table can have only one PRIMARY KEY whereas there can be any number of UNIQUE keys. The columns that compose PK are automatically define NOT NULL, whereas a column that compose a UNIQUE is not automatically defined to be mandatory must also specify the column is NOT NULL.

9. Can a primary key contain more than one columns

Yes

10. How you will avoid duplicating records in a query

By using DISTINCT

11. What is difference between SQL and SQL*PLUS

SQL*PLUS is a command line tool where as SQL and PL/SQL language interface and reporting tool. Its a command line tool that allows user to type SQL commands to be executed directly against an Oracle database. SQL is a language used to query the relational database(DML,DCL,DDL). SQL*PLUS commands are used to format query result, Set options, Edit SQL commands and PL/SQL.

12. Which datatype is used for storing graphics and images

LONG RAW data type is used for storing BLOB's (binary large objects).

13. How will you delete duplicating rows from a base table

DELETE FROM table_name A WHERE rowid>(SELECT min(rowid) from table_name B where B.table_no=A.table_no);

CREATE TABLE new_table AS SELECT DISTINCT * FROM old_table;

DROP old_table RENAME new_table TO old_table DELETE FROM table_name A WHERE rowid NOT IN (SELECT MAX(ROWID) FROM table_name GROUP BY column_name)

14. What is difference between SUBSTR and INSTR

SUBSTR returns a specified portion of a string eg SUBSTR('BCDEF',4) output BCDE INSTR provides character position in which a pattern is found in a string. eg INSTR('ABC-DC-F','-',2) output 7 (2nd occurence of '-')

17. When you use WHERE clause and when you use HAVING clause

HAVING clause is used when you want to specify a condition for a group function and it is written after GROUP BY clause The WHERE clause is used when you want to specify a condition for columns, single row functions except group functions and it is written before GROUP BY clause if it is used.

18. Which is more faster - IN or EXISTS

EXISTS is more faster than IN because EXISTS returns a Boolean value whereas IN returns a value.

19. What is a OUTER JOIN

Outer Join--Its a join condition used where you can query all the rows of one of the tables in the join condition even though they dont satisfy the join condition.

1. What is a pseudo column. Give some examples

It is a column that is not an actual column in the table.

eg USER, UID, SYSDATE, ROWNUM, ROWID, NULL, AND LEVEL.

2. What is the purpose of a cluster.

Oracle does not allow a user to specifically locate tables, since that is a part of the function of the RDBMS. However, for the purpose of increasing performance, oracle allows a developer to create a CLUSTER. A CLUSTER provides a means for storing data from different tables together for faster retrieval than if the table placement were left to the RDBMS.

3. What is a cursor.

Oracle uses work area to execute SQL statements and store processing information PL/SQL construct called a cursor lets you name a work area and access its stored information A cursor is a mechanism used to fetch more than one row in a Pl/SQl block.

4. Difference between an implicit & an explicit cursor.

PL/SQL declares a cursor implicitly for all SQL data manipulation statements, including quries that return only one row. However,queries that return more than one row you must declare an explicit cursor or use a cursor FOR loop. Explicit cursor is a cursor in which the cursor name is explicitly assigned to a SELECT statement via the CURSOR...IS statement. An implicit cursor is used for all SQL statements Declare, Open, Fetch, Close. An explicit cursors are used to process multirow SELECT statements An implicit cursor is used to process INSERT, UPDATE, DELETE and single row SELECT. .INTO statements. TRUE instead.

14. Can cursor variables be stored in PL/SQL tables.If yes how.If not why.

No, a cursor variable points a row which cannot be stored in a two-dimensional PL/SQL table.

15. Difference between procedure and function.

Functions are named PL/SQL blocks that return a value and can be called with arguments procedure a named block that can be called with parameter. A procedure all is a PL/SQL statement by itself, while a Function call is called as part of an expression.

16. What are different modes of parameters used in functions and procedures.

IN OUT INOUT

25. Can you use a commit statement within a database trigger.

No

7. Describe third normal form

Something like: In third normal form all attributes in an entity are related to the primary key and only to the primary key

9. What is an ERD

An ERD is an Entity-Relationship-Diagram. It is used to show the entities and relationships for a database logical model.

Joins:

A Join is the process of combining data from two or more tables. The DBMS takes all combinations of rows from the given tables.

Cross Join:

Join without filter conditions.

A Cross Join is the Cartesian product or the result of all possible combinations of the rows from each of the tables involved in the join operation. This occurs when, no specific Join conditions (filters) are specified.

For eg: there are 3 tables A,B and C with 10,20 and 30 number of rows respectively. So a cartesian production would happen in the below scenario,

Select A.col1, B.col2, C.col3 from A, B, C
--No where condition

which returns 10x20x30=6000 records are result.

Thus the number of rows in a Cartesian product of two tables is equal to the number of rows in the first table times the number of rows in the second table. Never use this kind of joins unless unavaidable as this takes huge amount of memory to sort and store

Natural Join: (ANSI Joins)

These are ANSI Joins which are used for portability. You can use this in almost all Standard Databases like Oracle, Microsoft SQL Server etc.

SELECT DNAME, ENAME, MGR FROM DEPARTMENTS NATURAL JOIN EMPLOYEES;

PS: Both tables should have primary key-referential key relationship.

Self Join:

Sometimes there are scenarios where we might need to do Join operations within the same table. Such joins which refers to the same, single table are known as a Self Joins

For Eg: If you need to get all employees and their managers, then you could do a self join.

SELECT E1.ENAME||' reports to '||E2.ENAME FROM EMP E1, EMP E2

WHERE E1.MGR = E2.EMPNO;

PS: This is just to show an example for Self Join. The results may not be accurate.

Inner Join or **Simple Join** or **Equi Join:**

Inner joins returns only the rows from the cross product that meets the join condition/s.

SELECT ENAME, JOB, DEPT.DEPTNO, DNAME FROM EMP, DEPT WHERE EMP.DEPTNO = DEPT.DEPTNO;

Returns data only from those records which matches the condition EMP.DEPTNO = DEPT.DEPTNO, from both tables

Outer Join:

They are of 2 types:

a) Left Outer Join

b) Right Outer Join

Suppose there are 2 tables A and B.

Left Outer Join

Returns all records from table A and only those matching with the join operation from Table B

For eg:

```
SELECT * FROM  EMP, DEPT
WHERE EMP.DEPTNO  = DEPT.DEPTNO (+);
```

Returns all records from EMP and only those records from DEPT which matches the condition EMP.DEPTNO = DEPT.DEPTNO

Right Outer Join

Returns all records from table B and only those matching with the join operation from Table A (just the reverse of left outer join)

For eg:

```
        SELECT * FROM    EMP, DEPT
        WHERE EMP.DEPTNO (+)  = DEPT.DEPTNO ;
```

Returns all records from DEPT and only those records from EMP which matches the condition EMP.DEPTNO = DEPT.DEPTNO

SQL Injection

1. Overview

The article describes the SQL injection attacks on applications. It demonstrates the code exploitation, ways of rectification and code designing recommendations.

2. Introduction

SQL injection refers to the attack on the applications which have Oracle as back end database. The activity to inject SQL and PL/SQL code through an application is SQL injection. It is a serious vulnerability and may lead to fatal consequences. An unauthorized attacker may get full database access by inputting bad input through application which may lead to leakage of important data.

3. A basic demonstration of SQL injection

The SQL query below gets the password of an input user from the USERS table.
Code sql:

```sql
SELECT USERNAME, PASSWORD FROM USERS  WHERE USERNAME =

'CLUB'
```

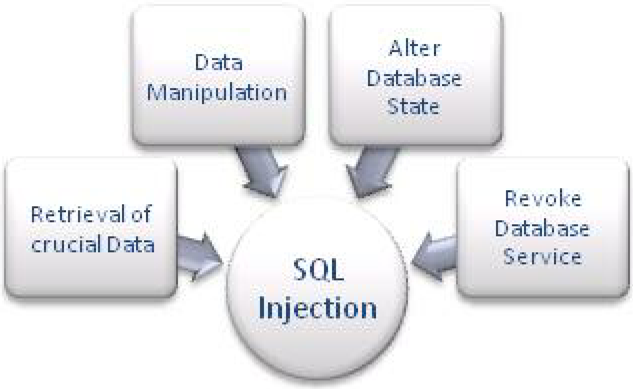
Developers are ignorant that their above query can be misused to an extent that it can list login names and passwords, which is relevant and crucial information for an organization. An invader can give input as
Code : "OR 1=1
which can retrieve all login and password.
Code sql:

```sql
SELECT USERNAME, PASSWORD
FROM USERS
WHERE USERNAME = ''
OR 1=1
```

In the above SQL, WHERE clause has two filter conditions. First condition yields FALSE while the second condition gives TRUE output. Since both are combined using OR logical operator, the combination gives TRUE and query retrieves all the login names and their passwords.

Like the above technique, SQL code can be exploited in multiple ways. Dynamic SQL and User/Definer right preserved subprograms are most prone to SQL injections.

4. Impact of SQL Injection



5. Categories of SQL Injection

Depending on the impact of SQL Injection, it can be divided into two major categories.

First Order Attack	**Second Order Attack**
Bad string Input (using quote and concatenation operator)	Performing an alternate activity to an ongoing system activity

6. SQL Injection: Example

A procedure P_GET_SAL was created to get the salary of input Employee Id.
Code sql:

```
CREATE OR REPLACE PROCEDURE P_GET_SAL  (P_ENAME VARCHAR2
DEFAULT NULL)
AS
CUR SYS_REFCURSOR;
V_ENAME VARCHAR2(100);
V_SAL NUMBER;
BEGIN
  V_STMT := 'SELECT ENAME, SALARY FROM EMPLOYEE  WHERE ENAME =
'''|| P_ENAME || '''';
  DBMS_OUTPUT.PUT_LINE(V_STMT);
  OPEN CUR FOR V_STMT;
  LOOP
    FETCH CUR INTO V_ENAME, V_SAL;
    EXIT WHEN CUR%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Employee : '||V_ENAME||' draws
'||TO_CHAR(V_SAL));
  END LOOP;
  CLOSE CUR;
END;
```

A malicious input can be given in below ways to inject the SQL. Illustration is as below.
Code sql:

```
SQL> EXEC P_GET_SAL(‘KING’);
Employee KING draws 4500

PL/SQL PROCEDURE successfully completed.
```

Code sql:

```
SQL> EXEC P_GET_SAL('KING'' UNION SELECT ENAME, SALARY FROM
EMPLOYEE WHERE 1=1');
Employee KING draws 4500
Employee ALLEN draws 1200
Employee MIKE draws 3400
Employee KATE draws 2300
Employee PAUL draws 6400
Employee TOMY draws 2700
Employee JENNY draws 6200
Employee JAKES draws 4600

PL/SQL PROCEDURE successfully completed.
```

7. Strategies to Resolve SQL Injection

Several strategies can be adopted to safeguard the SQL code and eradicate the impacts of SQL injection in applications. Some of them are listed below.

1. Use of Static SQL
2. Using Invoker’s rights
3. Use of Dynamic SQL with bind arguments
4. Validate and sanitize input using DBMS_ASSERT

2. Entity Relationship Model

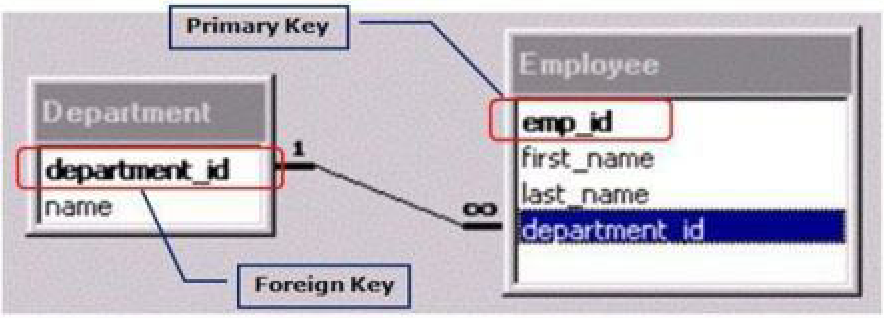
Entity Relationship Model is the one which is designed in Logical phase of DB design. It refers to the logical design of data as entity and its relationship with other entities. In technical terms, “entity” is a table which stores information, while “Relationship” is the relation between the entities.

A sample ER model

EMPLOYEE and DEPARTMENT are entities which store Employee and Department information in the tables. Consider few statements as below

Employee Id of an employee cannot be NULL or duplicate.
Every employee must belong to a Department.

Above two statements are enough to create an ER model for the data.



Relation database design stands on the fundamentals proposed by Mr. E.F. Codd. He identified 12 rules which a relational database must satisfy. Some references also include an additional rule which says that “A Relational Database Management System must be able to use it relational capabilities to manage the stored data”.

Codd’s rules of Database Design

Synopsis of the remaining rules are listed as below

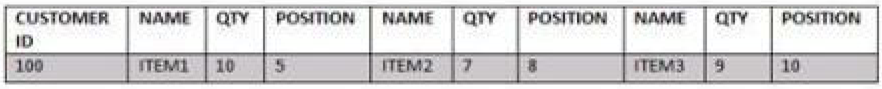
1. All information in the database must be stored as values in a table
2. All database information must be accessible through the combination of a table name, primary key and column name.
3. The database must use NULL values to indicate missing or unknown information
4. The database schema must be described using the relational database syntax
5. The database may support multiple languages, but it must support at least one language that provides full database functionality (e.g. SQL)
6. The system must be able to update all updatable views
7. The database must provide single-operation insert, update and delete functionality
8. Changes to the physical structure of the database must be transparent to applications and users.
9. Changes to the logical structure of the database must be transparent to applications and users.
10. The database must natively support integrity constraints.
11. Changes to the distribution of the database (centralized vs. distributed) must be transparent to applications and users.
12. Any languages supported by the database must not be able to subvert integrity controls.

3. Normalization

Data Normalization is the activity used to organize data in the database to sweep out redundant data and ensure sensible relationships. A normalized database not only ensures logical storage of data but also upgrades performance and data interactivity. A database can be normalized up to below degrees.

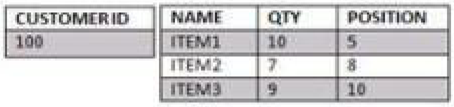
First Normal Form – Moving repetitive attributes in a table to a separate entity. Consider the below example of ORDER data for a day for a given Customer. Note that a Customer Id 100 orders 3 items in the day. In non-normalized state, all attribute values are stored in the same record. Note the iteration of the NAME, QTY, and POS attributes. In normalized state, the repeating attributes are segregated in separate records.

Before Normalization



CUSTOMER ID	NAME	QTY	POSITION	NAME	QTY	POSITION	NAME	QTY	POSITION
100	ITEM1	10	5	ITEM2	7	8	ITEM3	9	10

After Normalization



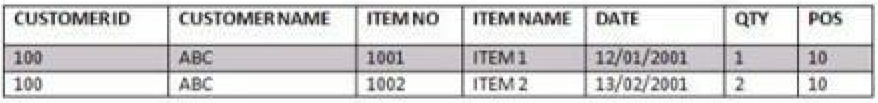
CUSTOMERID
100

NAME	QTY	POSITION
ITEM1	10	5
ITEM2	7	8
ITEM3	9	10

Second Normal Form – Once the data passes the 1NF, the attributes which are same in multiple rows can be moved to separate tables. 2NF talks about creating relationship between the two tables where one table contains the primary key data, while the other one contains the repetitive data for each key value.

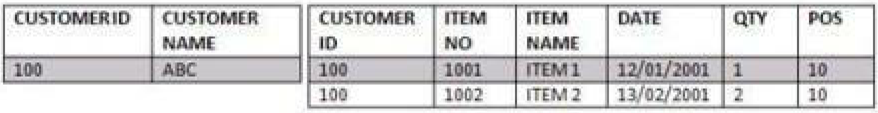
Consider an example where Customer data needs to be normalized. A record contains Customer Id as key, its details and Item details. Since items and customer share a relationship, they logically appear in multiple records. But, customer details, which do not share any relation with items, unnecessarily repeat in the records. They can be maintained in separate table.

Before Normalization



CUSTOMERID	CUSTOMERNAME	ITEMNO	ITEMNAME	DATE	QTY	POS
100	ABC	1001	ITEM 1	12/01/2001	1	10
100	ABC	1002	ITEM 2	13/02/2001	2	10

After Normalization



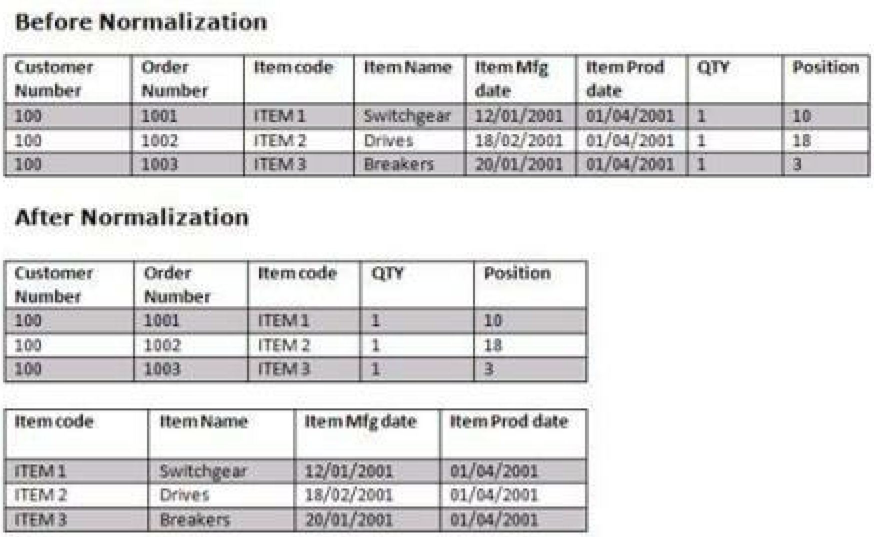
CUSTOMERID	CUSTOMER NAME
100	ABC

CUSTOMER ID	ITEM NO	ITEM NAME	DATE	QTY	POS
100	1001	ITEM 1	12/01/2001	1	10
100	1002	ITEM 2	13/02/2001	2	10

Third Normal Form

Once the data passes through 1NF and 2NF, it is eligible for 3NF. It refers to the removal of non key attributes. The attributes which are not dependent on the primary key of the table must be grouped as a separate entity.

In the below example, Customer and Items share relationship. The entity “Item” contains non key attributes like Item Mfg date and Item Prod date, which are not dependent on the Order Number Key. Instead, they describe a property of an Item.



Before Normalization

Customer Number	Order Number	Item code	Item Name	Item Mfg date	Item Prod date	QTY	Position
100	1001	ITEM 1	Switchgear	12/01/2001	01/04/2001	1	10
100	1002	ITEM 2	Drives	18/02/2001	01/04/2001	1	18
100	1003	ITEM 3	Breakers	20/01/2001	01/04/2001	1	3

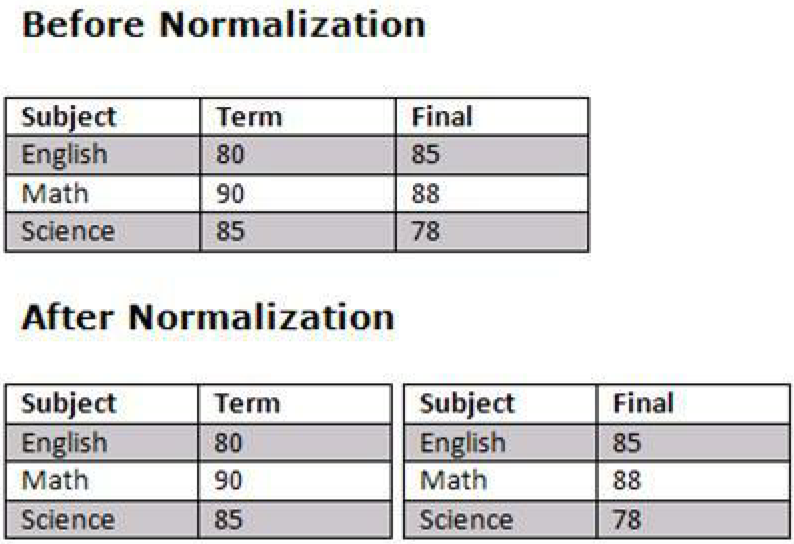
After Normalization

Customer Number	Order Number	Item code	QTY	Position
100	1001	ITEM 1	1	10
100	1002	ITEM 2	1	18
100	1003	ITEM 3	1	3

Item code	Item Name	Item Mfg date	Item Prod date
ITEM 1	Switchgear	12/01/2001	01/04/2001
ITEM 2	Drives	18/02/2001	01/04/2001
ITEM 3	Breakers	20/01/2001	01/04/2001

The normalization levels proposed by E.F. Codd are considered as standard techniques. Additionally, Boyce Codd Normal Form (BCNF) addresses dependencies between attributes and Candidate key. Candidate key is the column which is eligible to be a primary key. If candidate key becomes primary key, then 1NF, 2NF, and 3NF apply to BCNF classification also. Below are the additional normalization levels.

Fourth Normal Form: Multi-valued dependencies are broken down into separate entities. This implies that the attributes which are not dependent on the primary key can be segregated into different tables.
In the example, the table records the Marks obtained by a student in Term and Final exams in three subjects. Since Term and Final attributes are mutually exclusive and key column dependent, they can be treated as separate entities.



Before Normalization

Subject	Term	Final
English	80	85
Math	90	88
Science	85	78

After Normalization

Subject	Term
English	80
Math	90
Science	85

Subject	Final
English	85
Math	88
Science	78

Fifth Normal Form: It is also called projection-join Normal form. A data satisfies 5NF if a unique pair is formed by the combination of one or more candidate keys. This implies that a set of data can be represented in more than one small entity.

4. Additional terms used in ER Modeling

In addition to above fundamentals, below are some terms which are required to be familiar with during DB design process. Below are few of them

1. **Entity and Relationship** – The objects whose information is required to be stored is called as Entity. It represents a thing or an event. An entity has differentiable characteristics and optionally, it can have descriptive characteristics too.
An entity can be associated to another entity through a defined Relationship. In a statement “Oracle is best DBMS in the world”, the terms ‘Oracle’ and ‘DBMS’ are entities while ‘is best’ defines the relationship between the two. For example, ‘Employee’ is an entity.

Advantages of ER Model

- Easy to understand
- Eases the physical design from Logical design

Disadvantages

- The ER model is focused on Data and not on process or methodology.

2. **Attributes** – Every entity contains logical set of attributes which define a characteristic of the entity by storing a value. For an entity ‘Employee’, ‘Salary’ and ‘Department’ are attributes.
3. **Cardinality of Relationships** – Relationship between the entities (tables) can exist in three forms as listed below.
a. One-to-one relationship – Two entities are related by a single relationship. For example, a Manager can manage one and only one Department.
b. One-to-many relationship – An entity can share multiple relationships with the other entity. For example, a Department can contain multiple Employees.
c. Many-to-many relationship – Two entities can share multiple relationship with each other. For example, a Master Part can contain several Parts, while a Part can be used in multiple Master Parts.

4. **Primary Key** – It is a column in the table which carries a unique and definite value throughout the table.

5. **Foreign Key** – It is a column in the Child table which always points to an attribute value in its Parent table.