delete duplicate rows (solution)

We have a table with duplicate records for columnA, and we want to write a SQL query to make columnA the primary key. For this we need a delete query to delete the duplicate records so that columnA will have only unique values.

To remove duplicate rows of data, use the following statement:

DELETE FROM my\_table

WHERE rowid NOT IN ( SELECT MAX(ROWID) FROM my\_table

GROUP BY colA,colB,colC );

In the GROUP BY clause, enumerate all of your columns in your table, or the columns you think should be the primary key columns. The subquery will get the max rowid of these groupings. The DELETE will remove all rows that do not have these rowid values.

* Difference between delete, drop and truncate

**DELETE**  
  
1. DELETE is a DML statement.  
2. DELETE removes some rows if WHERE clause is used  
3. Can be rolled back  
4. Can be used with or without WHERE clause  
5. Does not reset identity of the table  
6. Triggers will be fired.  
7. When DELETE operation is performed, all the data get copied into Rollback Tablespace first,   
and then delete operation get performed. Hence we can get back the data by ROLLBACK command.  
  
**SYNTAX:**  
  
To delete a particular row  
  
DELETE FROM table\_name  
WHERE column\_name = column\_value  
  
To delete all rows  
  
DELETE FROM table\_name  
Or  
DELETE \* FROM table\_name  
  
**DROP**  
1. DROP is a DDL statement.  
2. Removes a table from the database. Table structures, indexes, privileges, constraints will   
also be removed.  
3. Cannot be rolled back  
4. No Triggers will be fired.  
  
**SYNTAX:**  
  
DROP TABLE table\_name  
  
  
**TRUNCATE**  
1. TRUNCATE is a DDL Statement.  
2. Removes all rows from a table, but the table structures and its columns, constraints, indexes   
remains.   
3. Cannot be rolled back  
4. Resets the identity of the table  
5. Truncate is faster and uses fewer system and transaction log than delete.  
6. Cannot use TRUNCATE on a table referenced by a FOREIGN KEY constraint.  
7. No Triggers will be fired.  
8. Cannot use WHERE conditions  
  
**SYNTAX:**  
TRUNCATE TABLE table\_name

* Difference between WHERE and HAVING, GROUP BY clause

The SQL **GROUP BY** clause is used in collaboration with the SELECT statement to arrange identical data into groups. The GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

Having clause is used to specify condition after Group By, preceeded by ORDER by. Having clause is filter on GROUP BY result.

Where clause is filter on select querry.

Example:

Consider the CUSTOMERS table is having the following records:

+----+----------+-----+-----------+----------+

| ID | NAME | AGE | ADDRESS | SALARY |

+----+----------+-----+-----------+----------+

| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |

| 2 | Khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+-----+-----------+----------+

If you want to know the total amount of salary on each customer, then GROUP BY query would be as follows:

SQL> SELECT NAME, SUM(SALARY) FROM CUSTOMERS

GROUP BY NAME;

This would produce the following result:

+----------+-------------+

| NAME | SUM(SALARY) |

+----------+-------------+

| Chaitali | 6500.00 |

| Hardik | 8500.00 |

| kaushik | 2000.00 |

| Khilan | 1500.00 |

| Komal | 4500.00 |

| Muffy | 10000.00 |

| Ramesh | 2000.00 |

Order by is used to arrange data in sorted form.

* **INTERSECT Clause**: is used to combine two SELECT statements, but returns rows only from the first SELECT statement that are identical to a row in the second SELECT statement.
* **Except Clause:** combines two SELECT statements and returns rows from the first SELECT statement that are not returned by the second SELECT statement.
* **UNION clause:** Combines the results of two select statements and returns records without any duplicates.
* **UNION ALL clause:** Combines the results of two select statements and returns records along with duplicates.
* **CARTESIAN Join or CROSS Join:**
* SQL query **to find duplicate records** in a table along with count

SELECT FirstName, LastName, MobileNo, COUNT (1) AS CNT

FROM coustomers

GROUP BY FirstName, LastName, MobileNo

HAVING COUNT (1) > 1;

* SQL query to find 2nd largest salary

SQL query to find 2nd largest salary

SELECT MAX (SALARY) from employees

WHERE SALARY NOT IN ( SELECT MAX (SALARY) from employees) ;

OR using co-related subquery

Simple logic is if you wanna find 2nd largest salary there will be 1 salary higher than it. If you wanna find 3rd largest salary there will be 2 salaries higher than it.

2nd largest salary

SELECT \*

FROM Employee Emp1

WHERE (1) = (

SELECT COUNT(DISTINCT(Emp2.Salary))

FROM Employee Emp2

WHERE Emp2.Salary > Emp1.Salary)

3rd largest salary

SELECT \* from employee emp1

WHERE (2) =

(SELECT COUNT(DISTINCT(SALARY.emp2))

from employee emp2

WHERE emp2.SALARY > emp1.SALARY);

* SQL Query to join three tables

SELECT s.coulmn1, s.cloumn2, s.column3, r.coulmn1

From Table\_name1 AS s

INNER JOIN Table\_name2 AS r

ON s.column1 = r.column1

INNER JOIN Table\_name3 AS q

ON r.column1 = q. column1;

* SQL Query **to find customers who have not ordered anything yet.**

SELECT C.ID from COSTOMER AS C WHERE C.ID NOT IN (SELECT O.CUSTOMER\_ID from ORDER O)

Other way:

SELECT C.Name FROM Customers C

LEFT JOIN Orders O ON C.Id = O.CustomerId

WHERE O.CustomerId is NULL

* SQL query to find department number, name and number of employees in each department from department and employees table.

SELECT d.depart\_num, d.depart\_name, COUNT (e.employee\_number) AS Total\_Emp

FROM department d, employees e

WHERE d.depart\_num = e.depart\_num

GROUP BY d.depart\_num, d.depart\_name

* **SQL query** to find employee name and their manager name from single table employee

SELECT e1.emp\_name as EMPLOY\_NAME, e2.emp\_name as REPORTS\_TO

From EMPLOYEE AS e1

INNER JOIN

EMPLOYEE AS e2

ON e1.manager\_id = e2.emp\_id;

* **SQL query** to take backup of a table

CREATE TABLE employee\_backup AS

SELECT \* from employee

* SQL query to find count of males and females in single query (from Gender coulmn)

SELECT COUNT (case when GENDER=’MALE’ then 1 end) AS male\_count,

COUNT (case when GENDER=’FEMALE’ then 1 end) AS female\_count,

COUNT (\*) AS total\_count

FROM student

Group by GENDER;

Queries based on date range

Delete duplicate rows

delete from EmpDup where EmpID in(select EmpID from EmpDup group by EmpId having

count(\*) >1)

**1. What is the difference between Procedure and Function ?**

Answer:-

1) Function must return a value and procedure need not.

2) Function can be used in SQL with some restrictions. Procedure cannot be called directly from SQL.

**2. What is the difference between DELETE and TRUNCATE ?**

Answer:-

1) DELETE is a DML command and TRUNCATE is a DDL command.

2) TRUNCATE re-set the memory blocks after execution and much faster than DELETE in most of the circumstances.

**3. What is the difference between PRIMARY KEY and UNIQUE KEY constraints ?**

Answer:-

1) UNIQUE KEY columns can have null values but PRIMARY KEY column cannot accept null values.

2) A table can have only one PRIMARY KEY column (composite primary key is ok) but many UNIQUE KEY columns allowed.

**4. What is Cartesian Product ?**

Answer:-

1) If two or more tables are joining without join condition will result into Cartesian products.

2) If table A has 2 rows and table B has 4 rows then Cartesian product between A and B will return 8 rows ( 2 multiply by 4 )

**5. What is a Transaction ?**

Answer:-

1) Transaction is a logical unit of work which will end in a consistent status

2) In Oracle we can control transactions using Save points, Commit, rollback etc

**6. What is a Package, why do we go for packages ?**

Answer:-

1) Package is a collection of one more program units having Procedure, Function etc

2) Depends on business logic, all related program units can be tied into a single package for better maintenance, readability etc

**7. What is triggers, where we use triggers ?**

Answer:-

1) Procedural code which automatically executes based on an event like insert,update, delete etc

2) Auditing, Logging etc

**8. How to do sql performance tuning ?**

Answer:-

1) Basic sql performance starts with EXPLAIN PLAN

2) Explain cost, index, access methods, plan etc

**9. What is the most challenging/difficult feature in Oracle ?**

Answer:-

1) This is the most tricky question, answer based on your experience.

2) Do not answer with a trivial problem, try something advanced feature

**10. What is a materialized view ?**

Answer:-

1) A materialized view, also called as snapshot,which contains the results of a query.

2) Generally used to create summary tables based on aggregations of a table data.

http://www.java67.com/2013/04/10-frequently-asked-sql-query-interview-questions-answers-database.html

=========================

**Table Name : Employee**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Employee\_id** | **First\_name** | **Last\_name** | **Salary** | **Joining\_date** | **Department** |
| 1 | John | Abraham | 1000000 | 01-JAN-13 12.00.00 AM | Banking |
| 2 | Michael | Clarke | 800000 | 01-JAN-13 12.00.00 AM | Insurance |
| 3 | Roy | Thomas | 700000 | 01-FEB-13 12.00.00 AM | Banking |
| 4 | Tom | Jose | 600000 | 01-FEB-13 12.00.00 AM | Insurance |
| 5 | Jerry | Pinto | 650000 | 01-FEB-13 12.00.00 AM | Insurance |
| 6 | Philip | Mathew | 750000 | 01-JAN-13 12.00.00 AM | Services |
| 7 | TestName1 | 123 | 650000 | 01-JAN-13 12.00.00 AM | Services |
| 8 | TestName2 | Lname% | 600000 | 01-FEB-13 12.00.00 AM | Insurance |

**Table Name : Incentives**

|  |  |  |
| --- | --- | --- |
| **Employee\_ref\_id** | **Incentive\_date** | **Incentive\_amount** |
| 1 | 01-FEB-13 | 5000 |
| 2 | 01-FEB-13 | 3000 |
| 3 | 01-FEB-13 | 4000 |
| 1 | 01-JAN-13 | 4500 |
| 2 | 01-JAN-13 | 3500 |

SQL Queries Interview Questions and Answers on "SQL Select"

**1. Get all employee details from the employee table**

Select \* from employee

**2. Get First\_Name,Last\_Name from employee table**

Select first\_name, Last\_Name from employee

**3. Get First\_Name from employee table using alias name “Employee Name”**

Select first\_name Employee Name from employee

**4. Get First\_Name from employee table in upper case**

Select upper(FIRST\_NAME) from EMPLOYEE

**5. Get First\_Name from employee table in lower case**

Select lower(FIRST\_NAME) from EMPLOYEE

**6. Get unique DEPARTMENT from employee table**

select distinct DEPARTMENT from EMPLOYEE

**Don't Miss** - [SQL and Database theory Interview Questions](http://a4academics.com/interview-questions/53-database-and-sql/411-sql-interview-questions-and-answers-database)

**7. Select first 3 characters of FIRST\_NAME from EMPLOYEE**

**Oracle Equivalent of SQL Server SUBSTRING is SUBSTR**, Query : select substr(FIRST\_NAME,0,3) from employee  
 **SQL Server Equivalent of Oracle SUBSTR is SUBSTRING**, Query : select substring(FIRST\_NAME,1,3) from employee  
 **MySQL Server Equivalent of Oracle SUBSTR is SUBSTRING**. In MySQL start position is 1, Query : select substring(FIRST\_NAME,1,3) from employee

**8. Get position of 'o' in name 'John' from employee table**

**Oracle Equivalent of SQL Server CHARINDEX is INSTR**, Query : Select instr(FIRST\_NAME,'o') from employee where first\_name='John'  
 **SQL Server Equivalent of Oracle INSTR is CHARINDEX**, Query: Select CHARINDEX('o',FIRST\_NAME,0) from employee where first\_name='John'  
 **MySQL Server Equivalent of Oracle INSTR is LOCATE**, Query: Select LOCATE('o',FIRST\_NAME) from employee where first\_name='John'

**9. Get FIRST\_NAME from employee table after removing white spaces from right side**

select RTRIM(FIRST\_NAME) from employee

**10. Get FIRST\_NAME from employee table after removing white spaces from left side**

select LTRIM(FIRST\_NAME) from employee

**11. Get length of FIRST\_NAME from employee table**

**Oracle,MYSQL Equivalent of SQL Server Len is Length** , Query :select length(FIRST\_NAME) from employee  
 **SQL Server Equivalent of Oracle,MYSQL Length is Len**, Query :select len(FIRST\_NAME) from employee

**12. Get First\_Name from employee table after replacing 'o' with '$'**

select REPLACE(FIRST\_NAME,'o','$') from employee

**13. Get First\_Name and Last\_Name as single column from employee table separated by a '\_'**

**Oracle Equivalent of MySQL concat is '||'**, Query : Select FIRST\_NAME|| '\_' ||LAST\_NAME from EMPLOYEE  
 **SQL Server Equivalent of MySQL concat is '+'**, Query : Select FIRST\_NAME + '\_' +LAST\_NAME from EMPLOYEE  
 **MySQL Equivalent of Oracle '||' is concat**, Query : Select concat(FIRST\_NAME,'\_',LAST\_NAME) from EMPLOYEE

**14. Get FIRST\_NAME ,Joining year,Joining Month and Joining Date from employee table**

**SQL Queries in Oracle**, Select FIRST\_NAME, to\_char(joining\_date,'YYYY') JoinYear , to\_char(joining\_date,'Mon'), to\_char(joining\_date,'dd') from EMPLOYEE  
 **SQL Queries in SQL Server**, select SUBSTRING (convert(varchar,joining\_date,103),7,4) , SUBSTRING (convert(varchar,joining\_date,100),1,3) , SUBSTRING (convert(varchar,joining\_date,100),5,2) from EMPLOYEE  
 **SQL Queries in MySQL**, select year(joining\_date),month(joining\_date), DAY(joining\_date) from EMPLOYEE

"SQL Order By" Interview Questions

**15. Get all employee details from the employee table order by First\_Name Ascending**

Select \* from employee order by FIRST\_NAME asc

**16. Get all employee details from the employee table order by First\_Name descending**

Select \* from employee order by FIRST\_NAME desc

**17. Get all employee details from the employee table order by First\_Name Ascending and Salary descending**

Select \* from employee order by FIRST\_NAME asc,SALARY desc

"SQL Where Condition" Interview Questions

**18. Get employee details from employee table whose employee name is “John”**

Select \* from EMPLOYEE where FIRST\_NAME='John'

**19. Get employee details from employee table whose employee name are “John” and “Roy”**

Select \* from EMPLOYEE where FIRST\_NAME in ('John','Roy')

**20. Get employee details from employee table whose employee name are not “John” and “Roy”**

Select \* from EMPLOYEE where FIRST\_NAME not in ('John','Roy')

"SQL Wild Card Search" Interview Questions

**21. Get employee details from employee table whose first name starts with 'J'**

Select \* from EMPLOYEE where FIRST\_NAME like 'J%'

**22. Get employee details from employee table whose first name contains 'o'**

Select \* from EMPLOYEE where FIRST\_NAME like '%o%'

**23. Get employee details from employee table whose first name ends with 'n'**

Select \* from EMPLOYEE where FIRST\_NAME like '%n'

"SQL Pattern Matching" Interview Questions

**24. Get employee details from employee table whose first name ends with 'n' and name contains 4 letters**

Select \* from EMPLOYEE where FIRST\_NAME like '\_\_\_n' (Underscores)

**25. Get employee details from employee table whose first name starts with 'J' and name contains 4 letters**

Select \* from EMPLOYEE where FIRST\_NAME like 'J\_\_\_' (Underscores)

**26. Get employee details from employee table whose Salary greater than 600000**

Select \* from EMPLOYEE where Salary >600000

**27. Get employee details from employee table whose Salary less than 800000**

Select \* from EMPLOYEE where Salary <800000

**28. Get employee details from employee table whose Salary between 500000 and 800000**

Select \* from EMPLOYEE where Salary between 500000 and 800000

**29. Get employee details from employee table whose name is 'John' and 'Michael'**

Select \* from EMPLOYEE where FIRST\_NAME in ('John','Michael')

Interview Questions on "SQL DATE Functions"

**30. Get employee details from employee table whose joining year is “2013”**

**SQL Queries in Oracle**, Select \* from EMPLOYEE where to\_char(joining\_date,'YYYY')='2013'  
  
**SQL Queries in SQL Server**, Select \* from EMPLOYEE where SUBSTRING(convert(varchar,joining\_date,103),7,4)='2013'  
  
**SQL Queries in MySQL**, Select \* from EMPLOYEE where year(joining\_date)='2013'

**31. Get employee details from employee table whose joining month is “January”**

**SQL Queries in Oracle**, Select \* from EMPLOYEE where to\_char(joining\_date,'MM')='01' or Select \* from EMPLOYEE where to\_char(joining\_date,'Mon')='Jan'  
  
**SQL Queries in SQL Server**, Select \* from EMPLOYEE where SUBSTRING(convert(varchar,joining\_date,100),1,3)='Jan'  
  
**SQL Queries in MySQL**, Select \* from EMPLOYEE where month(joining\_date)='01'

**32. Get employee details from employee table who joined before January 1st 2013**

**SQL Queries in Oracle**, Select \* from EMPLOYEE where JOINING\_DATE <to\_date('01/01/2013','dd/mm/yyyy')  
  
**SQL Queries in SQL Server** (Format - “MM/DD/YYYY”), Select \* from EMPLOYEE where joining\_date <'01/01/2013'  
  
**SQL Queries in MySQL** (Format - “YYYY-DD-MM”), Select \* from EMPLOYEE where joining\_date <'2013-01-01'

**33. Get employee details from employee table who joined after January 31st**

**SQL Queries in Oracle**, Select \* from EMPLOYEE where JOINING\_DATE >to\_date('31/01/2013','dd/mm/yyyy')  
  
**SQL Queries in SQL Server and MySQL** (Format - “MM/DD/YYYY”), Select \* from EMPLOYEE where joining\_date >'01/31/2013'  
  
**SQL Queries in MySQL** (Format - “YYYY-DD-MM”), Select \* from EMPLOYEE where joining\_date >'2013-01-31'

**35. Get Joining Date and Time from employee table**

**SQL Queries in Oracle**, select to\_char(JOINING\_DATE,'dd/mm/yyyy hh:mi:ss') from EMPLOYEE  
  
**SQL Queries in SQL Server**, Select convert(varchar(19),joining\_date,121) from EMPLOYEE  
  
**SQL Queries in MySQL**, Select CONVERT(DATE\_FORMAT(joining\_date,'%Y-%m-%d-%H:%i:00'),DATETIME) from EMPLOYEE

**36. Get Joining Date,Time including milliseconds from employee table**

**SQL Queries in Oracle**, select to\_char(JOINING\_DATE,'dd/mm/yyyy HH:mi:ss.ff') from EMPLOYEE . Column Data Type should be “TimeStamp”  
  
**SQL Queries in SQL Server**, select convert(varchar,joining\_date,121) from EMPLOYEE  
  
**SQL Queries in MySQL**, Select MICROSECOND(joining\_date) from EMPLOYEE

**37. Get difference between JOINING\_DATE and INCENTIVE\_DATE from employee and incentives table**

Select FIRST\_NAME,INCENTIVE\_DATE - JOINING\_DATE from employee a inner join incentives B on A.EMPLOYEE\_ID=B.EMPLOYEE\_REF\_ID

**38. Get database date**

**SQL Queries in Oracle**, select sysdate from dual  
  
**SQL Queries in SQL Server**, select getdate()  
  
**SQL Query in MySQL**, select now()

"SQL Escape Characters" Interview Questions

**39. Get names of employees from employee table who has '%' in Last\_Name. Tip : Escape character for special characters in a query.**

**SQL Queries in Oracle**, Select FIRST\_NAME from employee where Last\_Name like '%?%%'

**SQL Queries in SQL Server**, Select FIRST\_NAME from employee where Last\_Name like '%[%]%'

**SQL Queries in MySQL**, Select FIRST\_NAME from employee where Last\_Name like '%\%%'

**40. Get Last Name from employee table after replacing special character with white space**

**SQL Queries in Oracle**, Select translate(LAST\_NAME,'%',' ') from employee  
  
**SQL Queries in SQL Server and MySQL**, Select REPLACE(LAST\_NAME,'%',' ') from employee

"SQL Group By Query" Interview Questions and Answers

**41. Get department,total salary with respect to a department from employee table.**

Select DEPARTMENT,sum(SALARY) Total\_Salary from employee group by department

**42. Get department,total salary with respect to a department from employee table order by total salary descending**

Select DEPARTMENT,sum(SALARY) Total\_Salary from employee group by DEPARTMENT order by Total\_Salary descending

SQL Queries Interview Questions and Answers on "SQL Mathematical Operations using Group By"

**43. Get department,no of employees in a department,total salary with respect to a department from employee table order by total salary descending**

Select DEPARTMENT,count(FIRST\_NAME),sum(SALARY) Total\_Salary from employee group by DEPARTMENT order by Total\_Salary descending

**44. Get department wise average salary from employee table order by salary ascending**

select DEPARTMENT,avg(SALARY) AvgSalary from employee group by DEPARTMENT order by AvgSalary asc

**45. Get department wise maximum salary from employee table order by salary ascending**

select DEPARTMENT,max(SALARY) MaxSalary from employee group by DEPARTMENT order by MaxSalary asc

**46. Get department wise minimum salary from employee table order by salary ascending**

select DEPARTMENT,min(SALARY) MinSalary from employee group by DEPARTMENT order by MinSalary asc

**47. Select no of employees joined with respect to year and month from employee table**

**SQL Queries in Oracle**, select to\_char (JOINING\_DATE,'YYYY') Join\_Year,to\_char (JOINING\_DATE,'MM') Join\_Month,count(\*) Total\_Emp from employee group by to\_char (JOINING\_DATE,'YYYY'),to\_char(JOINING\_DATE,'MM')  
  
**SQL Queries in SQL Server**, select datepart (YYYY,JOINING\_DATE) Join\_Year,datepart (MM,JOINING\_DATE) Join\_Month,count(\*) Total\_Emp from employee group by datepart(YYYY,JOINING\_DATE), datepart(MM,JOINING\_DATE)  
  
**SQL Queries in MySQL**, select year (JOINING\_DATE) Join\_Year,month (JOINING\_DATE) Join\_Month,count(\*) Total\_Emp from employee group by year(JOINING\_DATE), month(JOINING\_DATE)

**48. Select department,total salary with respect to a department from employee table where total salary greater than 800000 order by Total\_Salary descending**

Select DEPARTMENT,sum(SALARY) Total\_Salary from employee group by DEPARTMENT having sum(SALARY) >800000 order by Total\_Salary desc

Advanced SQL Queries Interview Questions and Answers

**49. Select employee details from employee table if data exists in incentive table ?**

select \* from EMPLOYEE where exists (select \* from INCENTIVES)

**Explanation** : Here "exists" statement helps us to do the job of If statement. Main query will get executed if the sub query returns at least one row. So we can consider the sub query as "If condition" and the main query as "code block" inside the If condition. We can use any SQL commands (Joins, Group By , having etc) in sub query. This command will be useful in queries which need to detect an event and do some activity.

**50. How to fetch data that are common in two query results ?**

select \* from EMPLOYEE where EMPLOYEE\_ID INTERSECT select \* from EMPLOYEE where EMPLOYEE\_ID < 4

**Explanation** : Here "INTERSECT" command is used to fetch data that are common in 2 queries. In this example, we had taken EMPLOYEE table in both the queries.We can apply INTERSECT command on different tables. The result of the above query will return employee details of "ROY" because, employee id of ROY is 3, and both query results have the information about ROY.

**51. Get Employee ID's of those employees who didn't receive incentives without using sub query ?**

select EMPLOYEE\_ID from EMPLOYEE  
MINUS  
select EMPLOYEE\_REF\_ID from INCENTIVES

**Explanation** : To filter out certain information we use MINUS command. What MINUS Command odes is that, it returns all the results from the first query, that are not part of the second query. In our example, first three employees received the incentives. So query will return employee id's 4 to 8.

**52. Select 20 % of salary from John , 10% of Salary for Roy and for other 15 % of salary from employee table**

SELECT FIRST\_NAME, CASE FIRST\_NAME WHEN 'John' THEN SALARY \* .2 WHEN 'Roy' THEN SALARY \* .10 ELSE SALARY \* .15 END "Deduced\_Amount" FROM EMPLOYEE

**Explanation** : Here, we are using "SQL CASE" statement to achieve the desired results. After case statement, we had to specify the column on which filtering is applied. In our case it is "FIRST\_NAME". And in then condition, specify the name of filter like John, Roy etc. To handle conditions outside our filter, use else block where every one other than John and Roy enters.

**53. Select Banking as 'Bank Dept', Insurance as 'Insurance Dept' and Services as 'Services Dept' from employee table**

SQL Queries in Oracle, SELECT distinct DECODE (DEPARTMENT, 'Banking', 'Bank Dept', 'Insurance', 'Insurance Dept', 'Services', 'Services Dept') FROM EMPLOYEE  
SQL Queries in SQL Server and MySQL, SELECT case DEPARTMENT when 'Banking' then 'Bank Dept' when 'Insurance' then 'Insurance Dept' when 'Services' then 'Services Dept' end FROM EMPLOYEE

**Explanation** : Here "DECODE" keyword is used to specify the alias name. In oracle we had specify, Column Name followed by Actual Name and Alias Name as arguments. In SQL Server and MySQL, we can use the earlier switch case statements for alias names.

**54. Delete employee data from employee table who got incentives in incentive table**

delete from EMPLOYEE where EMPLOYEE\_ID in (select EMPLOYEE\_REF\_ID from INCENTIVES)

**Explanation** : Trick about this question is that we can't delete data from a table based on some condition in another table by joining them. Here to delete multiple entries from EMPLOYEE table, we need to use Subquery. Entries will get deleted based on the result of Subquery.

**55. Insert into employee table Last Name with " ' " (Single Quote - Special Character)**

Tip - Use another single quote before special character  
Insert into employee (LAST\_NAME) values ('Test''')

**56. Select Last Name from employee table which contain only numbers**

Select \* from EMPLOYEE where lower(LAST\_NAME)=upper(LAST\_NAME)

**Explanation** : In order to achieve the desired result, we use "ASCII" property of the database. If we get results for a column using Lower and Upper commands, ASCII of both results will be same for numbers. If there is any alphabets in the column, results will differ.

**57. Write a query to rank employees based on their incentives for a month**

select FIRST\_NAME,INCENTIVE\_AMOUNT,DENSE\_RANK() OVER (PARTITION BY INCENTIVE\_DATE ORDER BY INCENTIVE\_AMOUNT DESC) AS Rank from EMPLOYEE a, INCENTIVES b where a.EMPLOYEE\_ID=b.EMPLOYEE\_REF\_ID

**Explanation** : In order to rank employees based on their rank for a month, "DENSE\_RANK" keyword is used. Here partition by keyword helps us to sort the column with which filtering is done. Rank is provided to the column specified in the order by statement. The above query ranks employees with respect to their incentives for a given month.

**58. Update incentive table where employee name is 'John'**

update INCENTIVES set INCENTIVE\_AMOUNT='9000' where EMPLOYEE\_REF\_ID=(select EMPLOYEE\_ID from EMPLOYEE where FIRST\_NAME='John' )

**Explanation** : We need to join Employee and Incentive Table for updating the incentive amount. But for update statement joining query wont work. We need to use sub query to update the data in the incentive table. SQL Query is as shown below.

"SQL Join" Interview Questions

**59. Select first\_name, incentive amount from employee and incentives table for those employees who have incentives**

Select FIRST\_NAME,INCENTIVE\_AMOUNT from employee a inner join incentives B on A.EMPLOYEE\_ID=B.EMPLOYEE\_REF\_ID

**60. Select first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000**

Select FIRST\_NAME,INCENTIVE\_AMOUNT from employee a inner join incentives B on A.EMPLOYEE\_ID=B.EMPLOYEE\_REF\_ID and INCENTIVE\_AMOUNT >3000

**61. Select first\_name, incentive amount from employee and incentives table for all employes even if they didn't get incentives**

Select FIRST\_NAME,INCENTIVE\_AMOUNT from employee a left join incentives B on A.EMPLOYEE\_ID=B.EMPLOYEE\_REF\_ID

**62. Select first\_name, incentive amount from employee and incentives table for all employees even if they didn't get incentives and set incentive amount as 0 for those employees who didn't get incentives.**

**SQL Queries in Oracle**, Select FIRST\_NAME,nvl(INCENTIVE\_AMOUNT,0) from employee a left join incentives B on A.EMPLOYEE\_ID=B.EMPLOYEE\_REF\_ID  
  
**SQL Queries in SQL Server**, Select FIRST\_NAME, ISNULL(INCENTIVE\_AMOUNT,0) from employee a left join incentives B on A.EMPLOYEE\_ID=B.EMPLOYEE\_REF\_ID  
  
**SQL Queries in MySQL**, Select FIRST\_NAME, IFNULL(INCENTIVE\_AMOUNT,0) from employee a left join incentives B on A.EMPLOYEE\_ID=B.EMPLOYEE\_REF\_ID

**63. Select first\_name, incentive amount from employee and incentives table for all employees who got incentives using left join**

**SQL Queries in Oracle**, Select FIRST\_NAME,nvl(INCENTIVE\_AMOUNT,0) from employee a right join incentives B on A.EMPLOYEE\_ID=B.EMPLOYEE\_REF\_ID  
  
**SQL Queries in SQL Server**, Select FIRST\_NAME, isnull(INCENTIVE\_AMOUNT,0) from employee a right join incentives B on A.EMPLOYEE\_ID=B.EMPLOYEE\_REF\_ID  
  
**SQL Queries in MySQL**, Select FIRST\_NAME, IFNULL(INCENTIVE\_AMOUNT,0) from employee a right join incentives B on A.EMPLOYEE\_ID=B.EMPLOYEE\_REF\_ID

**64. Select max incentive with respect to employee from employee and incentives table using sub query**

**SQL Queries in Oracle**, select DEPARTMENT,(select nvl(max(INCENTIVE\_AMOUNT),0) from INCENTIVES where EMPLOYEE\_REF\_ID=EMPLOYEE\_ID) Max\_incentive from EMPLOYEE  
  
**SQL Queries in SQL Server**, select DEPARTMENT,(select ISNULL(max(INCENTIVE\_AMOUNT),0) from INCENTIVES where EMPLOYEE\_REF\_ID=EMPLOYEE\_ID) Max\_incentive from EMPLOYEE  
  
**SQL Queries in SQL Server**, select DEPARTMENT,(select IFNULL (max(INCENTIVE\_AMOUNT),0) from INCENTIVES where EMPLOYEE\_REF\_ID=EMPLOYEE\_ID) Max\_incentive from EMPLOYEE

"Top N Salary" SQL Interview Questions and Answers

**65. Select TOP 2 salary from employee table**

**SQL Queries in Oracle**, select \* from (select \* from employee order by SALARY desc) where rownum <3  
  
**SQL Queries in SQL Server**, select top 2 \* from employee order by salary desc  
  
**SQL Queries in MySQL**, select \* from employee order by salary desc limit 2

**66. Select TOP N salary from employee table**

**SQL Queries in Oracle**, select \* from (select \* from employee order by SALARY desc) where rownum <N + 1  
  
**SQL Queries in SQL Server**, select top N \* from employee  
  
**SQL Queries in MySQL**, select \* from employee order by salary desc limit N

**67. Select 2nd Highest salary from employee table**

**SQL Queries in Oracle**, select min(salary) from (select \* from (select \* from employee order by SALARY desc) where rownum <3)  
  
**SQL Queries in SQL Server**, select min(SALARY) from (select top 2 \* from employee) a  
  
**SQL Queries in MySQL**, select min(SALARY) from (select \* from employee order by salary desc limit 2) a

**68. Select Nth Highest salary from employee table**

**SQL Queries in Oracle**, select min(salary) from (select \* from (select \* from employee order by SALARY desc) where rownum <N + 1)  
  
**SQL Queries in SQL Server**, select min(SALARY) from (select top N \* from employee) a  
  
**SQL Queries in MySQL**, select min(SALARY) from (select \* from employee order by salary desc limit N) a

"SQL Union" Query Interview Questions

**69. Select First\_Name,LAST\_NAME from employee table as separate rows**

select FIRST\_NAME from EMPLOYEE union select LAST\_NAME from EMPLOYEE

**70. What is the difference between UNION and UNION ALL ?**

Both UNION and UNION ALL is used to select information from structurally similar tables. That means corresponding columns specified in the union should have same data type. For example, in the above query, if FIRST\_NAME is DOUBLE and LAST\_NAME is STRING above query wont work. Since the data type of both the columns are VARCHAR, union is made possible. Difference between UNION and UNION ALL is that , UNION query return only distinct values.

SQL Interview Questions on "SQL Table Scripts"

**71. Write create table syntax for employee table**

Oracle -CREATE TABLE EMPLOYEE (  
EMPLOYEE\_ID NUMBER,  
FIRST\_NAME VARCHAR2(20 BYTE),  
LAST\_NAME VARCHAR2(20 BYTE),  
SALARY FLOAT(126),  
JOINING\_DATE TIMESTAMP (6) DEFAULT sysdate,  
DEPARTMENT VARCHAR2(30 BYTE) )  
SQL Server -CREATE TABLE EMPLOYEE(  
EMPLOYEE\_ID int NOT NULL,  
FIRST\_NAME varchar(50) NULL,  
LAST\_NAME varchar(50) NULL,  
SALARY decimal(18, 0) NULL,  
JOINING\_DATE datetime2(7) default getdate(),  
DEPARTMENT varchar(50) NULL)

**72. Write syntax to delete table employee**

DROP table employee;

**73. Write syntax to set EMPLOYEE\_ID as primary key in employee table**

ALTER TABLE EMPLOYEE add CONSTRAINT EMPLOYEE\_PK PRIMARY KEY(EMPLOYEE\_ID)

**74. Write syntax to set 2 fields(EMPLOYEE\_ID,FIRST\_NAME) as primary key in employee table**

ALTER TABLE EMPLOYEE add CONSTRAINT EMPLOYEE\_PK PRIMARY KEY(EMPLOYEE\_ID,FIRST\_NAME)

**75. Write syntax to drop primary key on employee table**

Alter TABLE EMPLOYEE drop CONSTRAINT EMPLOYEE\_PK;

**76. Write Sql Syntax to create EMPLOYEE\_REF\_ID in INCENTIVES table as foreign key with respect to EMPLOYEE\_ID in employee table**

ALTER TABLE INCENTIVES ADD CONSTRAINT INCENTIVES\_FK FOREIGN KEY (EMPLOYEE\_REF\_ID) REFERENCES EMPLOYEE(EMPLOYEE\_ID)

**77. Write SQL to drop foreign key on employee table**

ALTER TABLE INCENTIVES drop CONSTRAINT INCENTIVES\_FK;

**78. Write SQL to create Orcale Sequence**

CREATE SEQUENCE EMPLOYEE\_ID\_SEQ START WITH 0 NOMAXVALUE MINVALUE 0 NOCYCLE NOCACHE NOORDER;

**79. Write Sql syntax to create Oracle Trigger before insert of each row in employee table**

CREATE OR REPLACE TRIGGER EMPLOYEE\_ROW\_ID\_TRIGGER  
BEFORE INSERT ON EMPLOYEE FOR EACH ROW  
DECLARE  
seq\_no number(12);  
BEGIN  
select EMPLOYEE\_ID\_SEQ.nextval into seq\_no from dual ;  
:new EMPLOYEE\_ID :=seq\_no;  
END;  
SHOW ERRORS;

**80. Oracle Procedure81. Oracle View**

An example oracle view script is given below  
create view Employee\_Incentive as select FIRST\_NAME,max(INCENTIVE\_AMOUNT) INCENTIVE\_AMOUNT from EMPLOYEE a, INCENTIVES b where a.EMPLOYEE\_ID=b.EMPLOYEE\_REF\_ID group by FIRST\_NAME

**82. Oracle materialized view - Daily Auto Refresh**

CREATE MATERIALIZED VIEW Employee\_Incentive  
REFRESH COMPLETE  
START WITH SYSDATE  
NEXT SYSDATE + 1 AS  
select FIRST\_NAME,INCENTIVE\_DATE,INCENTIVE\_AMOUNT from EMPLOYEE a, INCENTIVES b   
where a.EMPLOYEE\_ID=b.EMPLOYEE\_REF\_ID

**83. Oracle materialized view - Fast Refresh on Commit**

Create materialized view log for fast refresh. Following materialized view script wont get executed if materialized view log doesn't exists  
  
CREATE MATERIALIZED VIEW MAT\_Employee\_Incentive\_Refresh  
BUILD IMMEDIATE  
REFRESH FAST ON COMMIT AS  
select FIRST\_NAME,max(INCENTIVE\_AMOUNT) from EMPLOYEE a, INCENTIVES b  
where a.EMPLOYEE\_ID=b.EMPLOYEE\_REF\_ID group by FIRST\_NAME

**84. What is SQL Injection ?**

SQL Injection is one of the the techniques uses by hackers to hack a website by injecting SQL commands in data fields.

============

1. **To fetch ALTERNATE records from a table. (EVEN NUMBERED)**select \* from emp where rowid in (select decode(mod(rownum,2),0,rowid, null) from emp);
2. **To select ALTERNATE records from a table. (ODD NUMBERED)**select \* from emp where rowid in (select decode(mod(rownum,2),0,null ,rowid) from emp);
3. **Find the 3rd MAX salary in the emp table.**select distinct sal from emp e1 where 3 = (select count(distinct sal) from emp e2 where e1.sal <= e2.sal);
4. **Find the 3rd MIN salary in the emp table.**select distinct sal from emp e1 where 3 = (select count(distinct sal) from emp e2where e1.sal >= e2.sal);
5. **Select FIRST n records from a table.**select \* from emp where rownum <= &n;
6. **Select LAST n records from a table**select \* from emp minus select \* from emp where rownum <= (select count(\*) - &n from emp);
7. **List dept no., Dept name for all the departments in which there are no employees in the department.**select \* from dept where deptno not in (select deptno from emp);    
   alternate solution:  select \* from dept a where not exists (select \* from emp b where a.deptno = b.deptno);  
   altertnate solution:  select empno,ename,b.deptno,dname from emp a, dept b where a.deptno(+) = b.deptno and empno is null;
8. **How to get 3 Max salaries ?**select distinct sal from emp a where 3 >= (select count(distinct sal) from emp b where a.sal <= b.sal) order by a.sal desc;
9. **How to get 3 Min salaries ?**select distinct sal from emp a  where 3 >= (select count(distinct sal) from emp b  where a.sal >= b.sal);
10. **How to get nth max salaries ?**  
    select distinct hiredate from emp a where &n =  (select count(distinct sal) from emp b where a.sal >= b.sal);
11. **Select DISTINCT RECORDS from emp table.**select \* from emp a where  rowid = (select max(rowid) from emp b where  a.empno=b.empno);
12. **How to delete duplicate rows in a table?**delete from emp a where rowid != (select max(rowid) from emp b where  a.empno=b.empno);
13. **Count of number of employees in  department  wise.**select count(EMPNO), b.deptno, dname from emp a, dept b  where a.deptno(+)=b.deptno  group by b.deptno,dname;
14. **Suppose there is annual salary information provided by emp table. How to fetch monthly salary of each and every employee?**

select ename,sal/12 as monthlysal from emp;

1. **Select all record from emp table where deptno =10 or 40.**

select \* from emp where deptno=30 or deptno=10;

1. **Select all record from emp table where deptno=30 and sal>1500.**

select \* from emp where deptno=30 and sal>1500;

1. **Select  all record  from emp where job not in SALESMAN  or CLERK.**

select \* from emp where job not in ('SALESMAN','CLERK');

1. **Select all record from emp where ename in 'BLAKE','SCOTT','KING'and'FORD'.**

select \* from emp where ename in('JONES','BLAKE','SCOTT','KING','FORD');

1. **Select all records where ename starts with ‘S’ and its lenth is 6 char.**

select \* from emp where ename like'S\_\_\_\_';

1. **Select all records where ename may be any no of  character but it should end with ‘R’.**

select \* from emp where ename like'%R';

1. **Count  MGR and their salary in emp table.**

select count(MGR),count(sal) from emp;

1. **In emp table add comm+sal as total sal  .**

select ename,(sal+nvl(comm,0)) as totalsal from emp;

1. **Select  any salary <3000 from emp table.**

select \* from emp  where sal> any(select sal from emp where sal<3000);

1. **Select  all salary <3000 from emp table.**

select \* from emp  where sal> all(select sal from emp where sal<3000);

1. **Select all the employee  group by deptno and sal in descending order.**

select ename,deptno,sal from emp order by deptno,sal desc;

1. **How can I create an empty table emp1 with same structure as emp?**

Create table emp1 as select \* from emp where 1=2;

1. **How to retrive record where sal between 1000 to 2000?**  
   Select \* from emp where sal>=1000 And  sal<2000
2. **Select all records where dept no of both emp and dept table matches.**  
   select \* from emp where exists(select \* from dept where emp.deptno=dept.deptno)
3. **If there are two tables emp1 and emp2, and both have common record. How can I fetch all the recods but common records only once?**  
   (Select \* from emp) Union (Select \* from emp1)
4. **How to fetch only common records from two tables emp and emp1?**  
   (Select \* from emp) Intersect (Select \* from emp1)
5. **How can I retrive all records of emp1 those should not present in emp2?**  
   (Select \* from emp) Minus (Select \* from emp1)
6. **Count the totalsa  deptno wise where more than 2 employees exist.**  
   SELECT  deptno, sum(sal) As totalsal  
   FROM emp  
   GROUP BY deptno  
   HAVING COUNT(empno) > 2

## Select Query

Page 1 of 3

[AddThis Social Bookmark Button](http://www.addthis.com/bookmark.php)

The SELECT query is the most often used query in any database. So , learning this query is very important. Even if you don't write it manually, as some tools make your task easier by providing code assists. But this query helps you to understand many aspects of database concepts, ranging from DBA task to PL/SQL. I hope this tutorial helps learning SELECT query faster and easier.

We will be referring  tables from default schema/user SCOTT in Oracle.

The generalized select query looks like :

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | SELECT   [ DISTINCT| ALL]                { \* |  [columnName  [ AS newColumnName ] ] ,                        [columnName1 [ AS newColumnName1]],                        .........,                        .........,                }               FROM tableName [ALIAS][,]               [WHERE <condition>]               [GROUP BY columnlist ] [HAVING <condition>]               [ORDER BY columnlist] |

We will start with simple query and then keep on building complex queries.

1. **Select all the details of the employee from emp table.**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp; |

1. **Select only the names of all the employees in emp table;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT ename FROM emp; |

1. **Select records using multiple  columns;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT eno,ename,sal FROM emp; |

1. **Select records Using alias;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT eno,ename,salary AS sal FROM emp; |

* 1. **Select records using multiple alias**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT eno AS empno,ename AS empname ,sal AS salary FROM emp; |

* 1. **Using functions to enhance alias**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT count(sal) as totalsal FROM emp; |

1. **Derived or Computed fields: Columns values were manipulated as it gets retrived.**
   1. Find the monthly salary of employee,( The salary stored is on Annum basis).

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT sal / 12 FROM emp; |

* 1. Using Alias to decorate derived or computed fields.

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT sal / 12 AS monthly\_salary FROM emp |

* 1. Calculate the sum of monthly salary and the commissions of the employee.

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT ename,(sal / 12) + nvl(comm,0) AS monthsalwithcomm FROM emp; |

1. **Using Aggregate functions: Count,Min,Max,Sum,Avg.**
   1. **Find the number of rows in emp ?**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT count(\*) FROM emp; |

* 1. **Find the how many different Job profile are there in Employee table ?**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT COUNT(DISTINCT job) FROM emp; |

* 1. **Find out how many people were given commision ? ( Count function does n't include null as it counts)**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT COUNT(comm)FROM emp; |



[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 |  |

**Q5:Select records  from two tables;**  
5.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT e.ename, d.deptno,e.sal FROM emp e,dept d; |

5.2 Select records from two tables depending condition suppose where deptno of from both emp and dept mathes

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | Select e.ename, d.deptno,e.sal from emp e,dept d  where e.deptno=d.deptno; |

**Q6:  Select records using where clause;**  
6.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT ename,sal FROM emp  WHERE sal>2000 |

6.2 Select records from emp and dept where both deptno matches

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT e.ename,e.sal,d.deptno FROM emp e,dept d WHERE e.deptno=d.deptno |

6.3 Select secords where empno=1003

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE empno=1003 |

6.4 Select records where ename is KING

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE ename='KING'; |

**Q7:Select records using or;**  
7.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT ename,sal FROM emp WHERE deptno=10 OR deptno= 20 |

7.2

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE city='LONDON' OR city='PARIS' |

7.3

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE sal>2000 OR comm IS NOT NULL |

**Q8: Select records using between and;**  
8.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE sal BETWEEN 2000 AND 3000 |

8.2

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM dept WHERE deptno BETWEEN 20 AND 40; |

8.3

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE empno BETWEEN 1001 TO 1010; |

**Q9: Select records using in;**  
9.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE ename IN('SCOTT','WARD','ALLEN') |

9.2

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT ename FROM emp WHERE sal IN(1000,2000.3000.3200,3300); |

**Q10: Select records using not in;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE ename NOT IN('SCOTT','WARD','ALLEN') |

**Q11: Select records using null;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE comm IS  NULL; |

**Q12:Select records using not null;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE comm IS NOT NULL; |

**Q13:Select records using like;**  
13.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE ename LIKE'S%'; |

13.2

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE ename LIKE 'sC\_\_\_'; |

13.3

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE ename LIKE 'AL%'; |

13.4

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE ename LIKE 'A\_L\_N'; |

13.5

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE ename LIKE '\_\_\_L%'; |

**Q14:Select records using not like;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE ename NOT LIKE 'A%'; |

**Q15:Select records using multiple conditions.**  
15.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE comm IS NOT NULL AND sal >2000 AND deptno>20; |

15.2

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE sal>2000 AND deptno>20 AND ename NOT LIKE(A%) |

15.3

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE deptno<20 AND comm IS NOT NULL OR sal IN{1000,2000,3000,4000}; |

**Q16:Select records using function;**  
16.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT ename,sum(sal+comm) AS total FROM emp |

16.2

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1  2 | SELECT count(\*) FROM emp;  16.3 |

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT deptno,avg(sal) AS avgsal FROM emp GROUP BY deptno; |

**Q17:Select records using order by;**  
17.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp ORDER BY sal |

17.2

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT ename,sal FROM emp ORDER BY sal,ename |

17.3

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT empno,ename,sal FROM emp ORDER BY sal DESC,empno,empname |

**Q18:Select records using group by;**  
18.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT deptno,sum(sal)FROM emp GROUP BY deptno |

18.2

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT deptno,sum(sal) FROM emp GROUP BY deptno,ename ,empno |

**Q19:Select records using group by and having clause;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT deptno,sum(sal) GROUP BY deptno HAVING deptno>20 |

**Q20:select with-in select;**  
20.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE max(sal)<(SELECT max(sal) FROM emp) |

20.2

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE deptno=(SELECT deptno FROM dept); |

20.3

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT \* FROM emp WHERE deptno NOT IN (SELECT deptno FROM dept); |

**Q21:create a table by selecting record from another table;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | CREATE TABLE emp2 AS SELECT \* FROM emp; |

**Q22:Select records using exist;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT deptno FROM dept d WHERE EXISTS (SELECT \* FROM emp e WHERE d.deptno = e.deptno); |

**Q23:Select sysdate;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT sysdate FROM dual; |

**Q24:Select constraint name, constraint\_type;**

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1  2  3 | SELECT constraint\_name, constraint\_type  FROM user\_constraints  WHERE table\_name = 'emp'; |

**Q25:Select  nextval, currval from sequence;**  
25.1

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT emp\_sequence.nextval FROM dual |

25.2

[?](http://www.bullraider.com/database/sql-tutorial/6-select-query)

|  |  |
| --- | --- |
| 1 | SELECT emp\_sequence.currval FROM dual |