SQL interview questions:

1. How to delete the duplicates from a table?

Input data (employee table)  
---------------  
CREATE TABLE EMPLOYEE (EID INT, FIRSTNAME VARCHAR(20),LASTNAME VARCHAR(20),PHONE INT, MAIL VARCHAR(30));

INSERT INTO EMPLOYEE VALUES(1,'Adam','Owens',11223344,'Adam@demo.com');

INSERT INTO EMPLOYEE VALUES(2,'Mark','Wills',22334455,'Mark@demo.com');

INSERT INTO EMPLOYEE VALUES(3,'Natasha','Lee',33445566,'natasha@demo.com');

INSERT INTO EMPLOYEE VALUES(4,'Adam','Owens',11223344,'Adam@demo.com');

INSERT INTO EMPLOYEE VALUES(5,'Riely','Jones',44556677,'riley@demo.com');

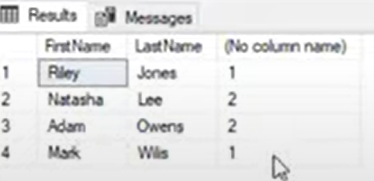
INSERT INTO EMPLOYEE VALUES(6,'Natasha','Lee',33445566,'natasha@demo.com');

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| empno | firstname | lastname | phone | email |
| 1 | Adam | Owens | 11223344 | [adam@demo.com](mailto:adam@demo.com) |
| 2 | Mark | Wills | 22334455 | [mark@demo.com](mailto:mark@demo.com) |
| 3 | Natasha | Lee | 33445566 | [natasha@demo.com](mailto:natasha@demo.com) |
| 4 | Adam | Owens | 11223344 | [adam@demo.com](mailto:adam@demo.com) |
| 5 | Riley | Jones | 44556677 | [riley@demo.com](mailto:riley@demo.com) |
| 6 | Natasha | Lee | 33445566 | [natasha@demo.com](mailto:natasha@demo.com) |

From the above table delete the duplicates and keep only one record.

***Step1) get the records count based on each group***

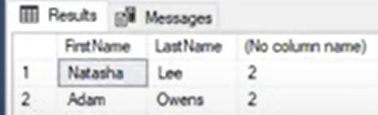
select firstname, lastname ,count(1) cnt from employee group by firstname,lastname.



Here we can see more than one record in the table.

***Step2) if need to find only duplicate record to find more than 1 record is:***

Select firstname,lastname,count(1) cnt from employee group by firstname,lastname having count(1)>1;



This is the how can be able to identify the duplicates.

***Step3) remove the duplicates from the repeated records, just retain the max value from the group.***

select max(employee\_id), firstname,lastname from employee group by firstname,lastname.

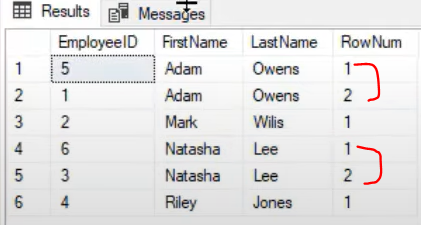
**To delete:**

Delete from employee where employee\_id   
not in (select max(employee\_id) from employee group by firstname,lastname

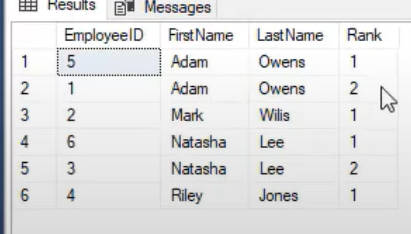
**Same query can be achieved by rownum() analytical fuction**

**Query)**

Select firstname,lastname,  
row\_number() over(partition firstname,lastname order by employee\_id) rownumber from employee.



**Same query can be write by Rank() function:**  
select \*,ROW\_NUMBER() over(partition by firstname,lastname order by eid) rownum from employee ;

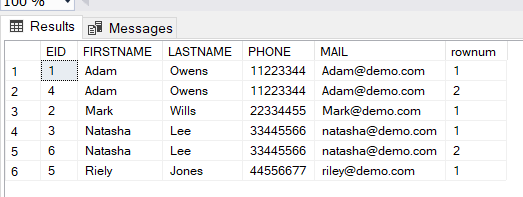


Same query can write by CTE( Common Table Expression):

with employee\_cte as (

select \*,ROW\_NUMBER() over(partition by firstname,lastname order by eid) rownum from employee )

select \* from employee\_cte



**Delete the duplicates based on the above query:**

with employee\_cte as (

select \*,ROW\_NUMBER() over(partition by firstname,lastname order by eid) rownum from employee )

delete from employee\_cte where rownum>1;

**How to find Nth Highest Salary?**

CREATE TABLE EMPLOYEE1 (EID INT, FIRSTNAME VARCHAR(20),LASTNAME VARCHAR(20),PHONE INT, MAIL VARCHAR(30),SALARY INT);

INSERT INTO EMPLOYEE1 VALUES(1,'Adam','Owens',11223344,'Adam@demo.com',6000);

INSERT INTO EMPLOYEE1 VALUES(2,'Mark','Wills',22334455,'Mark@demo.com',8500);

INSERT INTO EMPLOYEE1 VALUES(3,'Natasha','Lee',33445566,'natasha@demo.com',3000);

INSERT INTO EMPLOYEE1 VALUES(4,'Nick','Adams',77665544,'nick@demo.com',4500);

INSERT INTO EMPLOYEE1 VALUES(5,'Riely','Jones',44556677,'riley@demo.com',7500);

INSERT INTO EMPLOYEE1 VALUES(6,'Melessa','Rhodes',55443322,'melessa@demo.com',4000);

--waq to display whose salary is highest paid in the table

Select max(salary) From employee1;

--waq to display the employee all records of whose salary is highest paid

Select \* from employee1 where salary =(select max(salary) from employee1);

-- Display employee whose salary is lower than highest salary

Select \* from employee1 where salary < (select max(salary) from employee1);

-- Display employee whose salary is lower than highest salary order the salary in desc

Select \* from employee1 where salary < (select max(salary) from employee1) order by salary desc;

-- get the second highest salary

Select max(salary) sec\_hgstsal from employee1 where salary < (select max(salary) from employee1);

-- use the TOP command to get the top one records

Select top 1 \* from employee1 order by salary desc

-- use the TOP command to get the top 3 records

Select top 3 \* from employee1 order by salary desc

-- 5th highest salary

select top 1 \* from (

select top 5 \* from employee1 order by salary desc) sal\_order

order by salary ;

select \* from employee1 order by salary desc

--using the DENSE\_RANK Cand find the nth highest salary

select \*, DENSE\_RANK() over(order by salary desc) as sal\_order from employee1

-- find the nth salary using CTE

with sal\_order as (

select \*, DENSE\_RANK() over(order by salary desc) as sal\_order from employee1

)

select \* from sal\_order where sal\_order=4

 Employee Manager Hierarchy - Self Join - imp

CREATE TABLE EMPLOYEE2 (EID INT, FIRSTNAME VARCHAR(20),LASTNAME VARCHAR(20),PHONE INT, MAIL VARCHAR(30),SALARY INT,mgr\_id int);

INSERT INTO EMPLOYEE2 VALUES(1,'Adam','Owens',11223344,'Adam@demo.com',6000,3);

INSERT INTO EMPLOYEE2 VALUES(2,'Mark','Wills',22334455,'Mark@demo.com',8500,null);

INSERT INTO EMPLOYEE2 VALUES(3,'Natasha','Lee',33445566,'natasha@demo.com',3000,2);

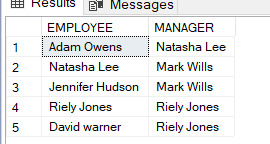
INSERT INTO EMPLOYEE2 VALUES(4,'Jennifer','Hudson',77665544,'jenni@demo.com',4500,2);

INSERT INTO EMPLOYEE2 VALUES(5,'Riely','Jones',44556677,'riley@demo.com',7500,5);

INSERT INTO EMPLOYEE2 VALUES(6,'David','warner',55443322,'david@demo.com',6500,5);

SELECT EMP.FIRSTNAME + ' ' + EMP.LASTNAME EMPLOYEE,

MGR.FIRSTNAME + ' ' + MGR.LASTNAME MANAGER FROM EMPLOYEE2 EMP INNER JOIN EMPLOYEE2 MGR ON EMP.MGR\_ID=MGR.EID;



# SQL Query - Convert data from Rows to Columns |Case | Pivot data

|  |  |  |
| --- | --- | --- |
| INPUT DATA | |  |
| **NAME** | **VALUE** | **ID** |
| NAME | ADAM | 1 |
| GENDER | MALE | 1 |
| SALARY | 1000 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| OUTPUT | | | |
| **ID** | **NAME** | **GENDER** | **SALARY** |
| 1 | ADAM | MALE | 1000 |

CREATE TABLE EMP\_PV(ID INT,VALUE VARCHAR(20),NAME VARCHAR(20));

INSERT INTO EMP\_PV VALUES(1,'ADAM','NAME');

INSERT INTO EMP\_PV VALUES(1,'MALE','GENDER');

INSERT INTO EMP\_PV VALUES(1,'1000','SALARY');

SELECT \* FROM EMP\_PV

SELECT

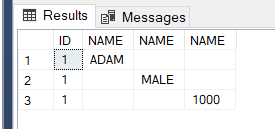
ID,

CASE WHEN NAME='NAME' THEN VALUE ELSE '' END AS NAME ,

CASE WHEN NAME='GENDER' THEN VALUE ELSE '' END AS NAME ,

CASE WHEN NAME='SALARY' THEN VALUE ELSE '' END AS NAME

FROM EMP\_PV;



SELECT

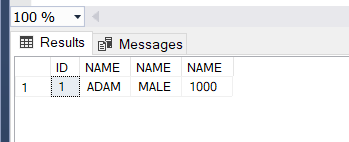
ID,

MAX(CASE WHEN NAME='NAME' THEN VALUE ELSE '' END) AS NAME ,

MAX(CASE WHEN NAME='GENDER' THEN VALUE ELSE '' END) AS NAME ,

MAX(CASE WHEN NAME='SALARY' THEN VALUE ELSE '' END) AS NAME

FROM EMP\_PV GROUP BY ID;



-- WE CAN ACHIEVE THIS BY USING **PIVOT** FUNCTION:

SELECT ID,[NAME],[GENDER],[SALARY]

FROM

(SELECT ID, NAME ENAME, VALUE FROM EMP\_PV) AS SOURCE\_TABLE

PIVOT

(MAX(VALUE)

FOR

ENAME IN ([NAME],[GENDER],[SALARY])

) AS PIVOT\_TABLE

(OR)

SELECT ID,NAME,GENDER,SALARY

FROM

(SELECT ID, NAME ENAME, VALUE FROM EMP\_PV) AS SOURCE\_TABLE

PIVOT

(MAX(VALUE)

FOR

ENAME IN (NAME,GENDER,SALARY)

) AS PIVOT\_TABLE

# Custom Sorting | Order by Month in an Year

CREATE TABLE SALES(MONTHSALES VARCHAR(20),SALES INT);

INSERT INTO SALES VALUES('APRIL',1200);

INSERT INTO SALES VALUES('MAY',4000);

INSERT INTO SALES VALUES('JUNE',2000);

INSERT INTO SALES VALUES('JANUARY',9000);

INSERT INTO SALES VALUES('FEB',1400);

INSERT INTO SALES VALUES('MARCH',3000);

INSERT INTO SALES VALUES('OCTOMBER',6000);

INSERT INTO SALES VALUES('NOVEMBER',1500);

INSERT INTO SALES VALUES('DECEMBER',9200);

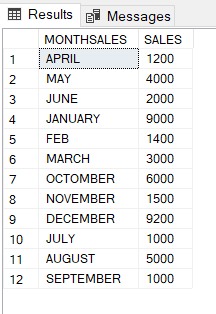
INSERT INTO SALES VALUES('JULY',1000);

INSERT INTO SALES VALUES('AUGUST',5000);

INSERT INTO SALES VALUES('SEPTEMBER',1000);

SELECT \* FROM SALES;

Query1) SELECT \* FROM SALES



Here data is not in order

Query2)  
SELECT \* FROM SALES

ORDER BY

CASE

WHEN MONTHSALES='JANUARY' THEN 1

WHEN MONTHSALES='FEB' THEN 2

WHEN MONTHSALES='MARCH' THEN 3

WHEN MONTHSALES='APRIL' THEN 4

WHEN MONTHSALES='MAY' THEN 5

WHEN MONTHSALES='JUNE' THEN 6

WHEN MONTHSALES='JULY' THEN 7

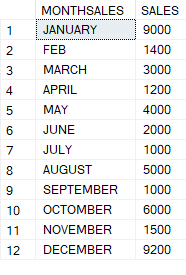
WHEN MONTHSALES='AUGUST' THEN 8

WHEN MONTHSALES='SEPTEMBER' THEN 9

WHEN MONTHSALES='OCTOMBER' THEN 10

WHEN MONTHSALES='NOVEMBER' THEN 11

WHEN MONTHSALES='DECEMBER' THEN 12 END ;



**Note**: if Date is also is in data, then if we need to sort the data based on the date of month then need to identify the month, based on month can sort the data.

# Compare with Previous Quarter's sales | Analytical Functions | Lead | Lag

CREATE TABLE QTR\_SALES( YR INT, QTRSALES VARCHAR(10),SALES INT);

INSERT INTO QTR\_SALES VALUES(2018,'Q1',5000);

INSERT INTO QTR\_SALES VALUES(2018,'Q2',5500);

INSERT INTO QTR\_SALES VALUES(2018,'Q3',2500);

INSERT INTO QTR\_SALES VALUES(2018,'Q4',10000);

INSERT INTO QTR\_SALES VALUES(2019,'Q1',10000);

INSERT INTO QTR\_SALES VALUES(2019,'Q2',5500);

INSERT INTO QTR\_SALES VALUES(2019,'Q3',3000);

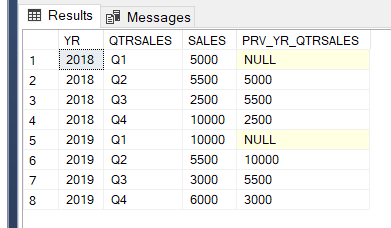
INSERT INTO QTR\_SALES VALUES(2019,'Q4',6000);

SELECT \* FROM QTR\_SALES;

SELECT YR,QTRSALES,SALES,

LAG(SALES) OVER (PARTITION BY YR ORDER BY QTRSALES) AS PRV\_YR\_QTRSALES

FROM QTR\_SALES;



BY using **Lead()** function also we can compare:

SELECT YR,QTRSALES,SALES,

LEAD(SALES) OVER (PARTITION BY YR ORDER BY QTRSALES) AS PRV\_YR\_QTRSALES

FROM QTR\_SALES;

# Split Concatenated String into Columns | CharIndex

Ex:

|  |
| --- |
| EMP\_NAME |
| Owena,Adam |
| Hopkins,David |
| Jones,Mary |
| Rhodes,Susan |

|  |  |  |
| --- | --- | --- |
| output |  |  |
| EMP\_NAME | FIRSTNAME | LASTNAME |
| Owena,Adam | Owena | Adam |
| Hopkins,David | Hopkins | David |
| Jones,Mary | Jones | Mary |
| Rhodes,Susan | Rhodes | Susan |

SELECT NAME,LEFT(NAME, CHARINDEX(',',NAME)-1) AS LAST\_NAME,

RIGHT(NAME,LEN(NAME)-CHARINDEX(',',NAME)) AS FIRSTNAME FROM EMP;

**Charindex**() – is similar to the ***instr()*** function in Oracle

# Replace special characters | Control Characters | REPLACE function

CREATE TABLE EMPLOYEE\_ADD (EID INT, FIRSTNAME VARCHAR(20),LASTNAME VARCHAR(20),PHONE INT, MAIL VARCHAR(30),SALARY INT,mgr\_id int,ADDRESS VARCHAR(70));

INSERT INTO EMPLOYEE\_ADD VALUES(1,'Adam','Owens',11223344,'Adam@demo.com',6000,3,'108 Main Street');

INSERT INTO EMPLOYEE\_ADD VALUES(2,'Mark','Wills',22334455,'Mark@demo.com',8500,null,'202 Ocean And Park Streat');

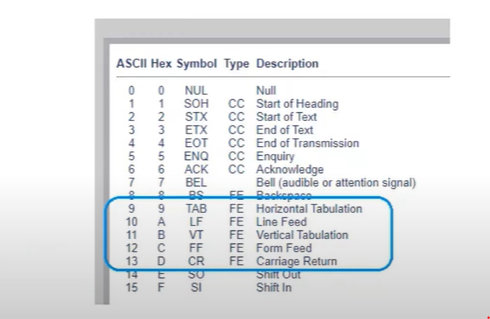
INSERT INTO EMPLOYEE\_ADD VALUES(3,'Natasha','Lee',33445566,'natasha@demo.com',3000,2,' 15 Roosvelt Rd ');

INSERT INTO EMPLOYEE\_ADD VALUES(4,'Jennifer','Hudson',77665544,'jenni@demo.com',4500,2,' 10 Main Street ');

INSERT INTO EMPLOYEE\_ADD VALUES(5,'Riely','Jones',44556677,'riley@demo.com',7500,5,' 909-1500 Victoria Street');

INSERT INTO EMPLOYEE\_ADD VALUES(6,'David','warner',55443322,'david@demo.com',6500,5,' 8754 New Market');

SELECT \* FROM EMPLOYEE\_ADD



SELECT \*, REPLACE(REPLACE(ADDRESS,CHAR(9),''),CHAR(13),'') FROM EMPLOYEE\_ADD;

**Date Functions:**

Use the below link for better date function practice in Sql Server.

<https://www.youtube.com/watch?v=tzfRdzgt1JM&list=PL2-GO-f-XvjBl5fpzdfYaPW28PwsLzLc4&index=11>

<https://www.youtube.com/watch?v=VqSD35pY9e4&list=PL2-GO-f-XvjBl5fpzdfYaPW28PwsLzLc4&index=12>

**For Alpha Numeric functions:**

<https://www.youtube.com/watch?v=zOiNdbFtLJU&list=PL2-GO-f-XvjBl5fpzdfYaPW28PwsLzLc4&index=13>

<https://www.youtube.com/watch?v=IKzGyfMgTBo&list=PL2-GO-f-XvjBl5fpzdfYaPW28PwsLzLc4&index=14>

<https://www.youtube.com/watch?v=yF4ku8aTaE0&list=PL2-GO-f-XvjBl5fpzdfYaPW28PwsLzLc4&index=15>

**Oracle Sql Interview questions:**

**What is data?**  
 it is a collection of rows and facts.

**What is Information?** information is nothing but meaningful data or process data. When we are processing the data then we are archiving meaningful result, this is called information.

Ex: student mark sheet, invoice of customers

**What is Data Retrieval?**

If we want to retrieve data from flat file then we must develop application programs in high level language, where as if we want to retrieve data from database then we are using SQL.

**What is Data Redundancy?**

. In Flat files sometimes we are maintaining multiple copy of the some data in different location.   
. this data is also called as ***duplicate data*** or ***redundant data*.**  
. if flat file whenever we are modifying this data is one location , it is not affected in other location. This is called ***inconsistency***, that’s why flat files doesn’t maintain consistency on data automatically.

. Databases automatically maintain consistent data through transactions every transactions internally having four properties; these are also called ACID property.

. If you want to reduce duplicate data in database then we are using “**Normalization Process**”

**What is Data Integrity?**Integrity means to maintain proper data. If want to maintain proper data in database then we are using constraints ( Primary key, foreign key,…) where as a flat files we must develop application program in high level language for maintaining valid data.

**What is Data Security?**Data stores in flat files can’t be secured because flat files does not provide security mechanism, where as database provide role based security.

**What is Data Indexing?**If we want to retrieve data very fast from database then we are using indexing mechanism on database. Whereas flat files don’t support indexing mechanism.

**How many types of SQL languages?**

1. **DDL (Data definition language)**
2. **DCL (Data Control Language)**
3. **DRL (Data Retrieval Language)**
4. **DML (Data Manipulation Language)**
5. **TCL (Transaction Control Language)**

**Difference between varchar and Varchar2?**

**Varchar2:** varchar2 datatype it is used to store variable length of alphanumeric data in bytes.

**Varchar**: prior to oracle 7.0 oracle having varchar datatype. It is also same as varchar2 datatype, varchar also stores variable length alphanumeric data in bytes but max size of the varchar datatype is upto 2000 types.

1. **Data Definition Language (DDL)?**

Create

Alter

Drop

Truncate

Rename

1. **Data Manipulation Language (DML)?**

Insert

Update

Delete

Merge

1. **Data Retrieval Language (DRL)**

Select

**What are Arithmetic Operators?**

+, \*, / , - (addition, multiplication, division, subtraction)

**What are Relational Operators?** =, <, >, =>, <=, <>, !=

**What are logical operators?** And   
 Or  
 Not

**What are Special Operators?**

1. IN Not IN
2. Between Not Between
3. ISNULL Is Not Null
4. Like Not Like

**What is NVL()?** NVL is a predefined function which are used to replace (or) Substitute use define value in place of Null

NVL(EXP1,EXP2);

**What is like Operator?**

This operation is used to search data based on character pattern like operator performance is very high compare to predefine searching function. Two special operator these re % and –

**Waq to display the employees whose ename start with M from emp table by using like operator?  
sql>** select \* from emp where ename like ‘M%’;  
  
**WAQ to display the employees whose ename having ‘M’ in any position within ename column from emp table?**

Sql> select \* from emp where ename like ‘%M%’?

**What is substr() function and example?**It will extract portion of the string within given string based on last two parameters

Ex:   
sql> select substr(‘abcdegh’,2,3) from dual;

Sql>select substr(‘abcdefgh’,-2,3) from dual;

**What is Instr() function?**instr() always returns number datatype ,that is returns position of the delimter, position of the character, position of the string within given string.

Sql> select instr(‘abcd\*d’,’\*’) from dual;  
 result: 4

Sql> select instr(‘ABCDEFGHCDIJKLCDMNP’,’CD’,-6,2) from dual;

Result: 3

Sql> select instr(‘ABCDEFGHCDIJKLCDMNP’,’CD’,-5,2) from dual;

Result: 9

**Lpad():**it will fills remaining spaces with specified character on the left side of given string.  
*syntax:* lpad (colname,total\_length,pad char) from dual;  
sql> select lpad(‘ABCD’,10,’#’) from dual;  
 result : ######ABCD

**RPAD():**sql> select rpad(‘abcd’,10,’#’) from dual;  
 result: abcd######

**Trim():**  
it is used to remove left and right side specified character , that is also used to remove leading and trailing spaces.  
*syntax:* trim(‘character’ from ‘string name’)

Sql> select trim(‘s’ from ‘ssthssmiss’) from dual;  
 result: thssmi

**Translate(), replace():**

*Truncate()* is used to replaces character by character where as *replace()* is used to replaces character by string.

Sql> select translate(‘india’,’in’,’xy’) from dual;  
 result: xydxa

Sql> select replace (‘india’,’in’,’xy’) from dual;

**Concat():**it is used to concatenate given two strings

Sql> select concat(‘wel’,’come’) from dual;  
 result: welcome

**Date functions:**

1. **Sysdate()**: It returns current date of the system in oracle date format

Syn: select sysdate from dual;

1. **Add\_months():** it is used to add or remove no.of months from the specified date based on 2nd parameters.

**Ex:** select add\_months(sysdate, 5) from dual;

1. **Last\_day():** it returns last date of the specified month.

**Ex:** select last\_day(sysdate) from dual;

**Joins:**

**Waq to display the employee who are working in the location ‘chicago’ from emp & dept table by using equi join?**

Sql> select ename,loc from emp,dept where emp.deptno=dept.deptno;

**Waq to display loc,no.of emploees, min.sal,max.sal from mep,dept table using equi joins?**

**Sql>** select loc,count(1),min(sal),max(sal) from emp e,dept d where e.deptno=d.deptno group by loc;

**Waq to display ename,mgrname from emp table by using selft join?**

**Sql>** select e1.ename employees, e2.ename “manager” from emp e1,emp e2 wehre e1.mgr=e2.empno;

**Waq to display the employees who are getting more salary than their manager salary from emp table by using self join?**

*Select e1.ename emplyees, e2.ename manager from emp e1, emp e2 where e1.mgr=e2.empno and e1.sal>e2.sal;*

**What is constraint:** constraints are used to prevent invalid data entry into tables.

List of constraints:

1. Not Null
2. Unique
3. Primary Key
4. Foreign key
5. Check
6. **waq to dispaly the employees who are working in SALES dept from emp, dept table?**

select \* from emp where deptno=(select deptno from emp where dname='SALES';

1. **waq to display senior most employee details from emp table.**

SELECT \* FROM EMP WHERE HIREDATE =(SELECT MIN(HIREDATE) FROM EMP);

SELECT \* FROM EMP WHERE HIREDATE ='17-DEC-80';

Scenarios:

----------

1. **waq to display the employees who are working same as 'SMITH' deptno from dept table?**

SELECT \* FROM EMP WHERE EMPNO =(SELECT EMPNO FROM EMP WHERE ENAME='SMITH'); - inner/child query returned only single record.

SELECT \* FROM EMP WHERE EMPNO IN (SELECT EMPNO FROM EMP WHERE ENAME='SMITH');

**VIEWS:**. View is a database object which is used to provides authority level or security.

. generally data security point of view, database administrator createing views from the table and then those views given to the no.of users.

. generally views doesn't' store data thats why view is also called as virtual table (or) window of a table.

. generally views are created from base table, based on the base table, views are categorized into two types.

**MATERIALIZED VIEWS:**

. Generally views doen't' stored data ,where as materialized views stores data.

. generatally materialized views are used to imporove the performance of the joins or aggiratable queries.

. materialized views stored result of the query .

. materialized views also stores data same like table but when are refreshing materialized views it synchronize the data based on base table.

**difference between views and m.views**

**views materialized views.**

1) View doesn't' store the data 1) materialized view stores data.

2) security purpose 2) improve performance purpose

3) when we are dropping base table then 3) when we are dropping base table also materialized view can can’t be accessible accessible

4) Through the view we can perform DML Operation 4) we can't' perform DML operation

**Q) if country getting the data from source, need to find the delim count**



**Query)**

select country

,regexp\_count(country, ',')+1 as tag\_count

from t;

<https://forums.oracle.com/ords/apexds/post/query-to-return-count-for-values-separated-by-commas-in-sin-0309>

**Q) find the similar who contain same salary in emp table?**

SELECT

e1.ename,e1.sal

FROM emp e1 INNER JOIN emp e2 ON e1.sal=e2.sal AND e1.ename<>e2.ename;

(OR)

SELECT \* FROM emp WHERE sal IN (  
SELECT sal FROM emp GROUP BY sal HAVING count(\*)>1) ;

**Q) find the last record from the table?**

SELECT \* FROM (SELECT \* FROM emp ORDER by rowid DESC) WHERE rownum<=1;

**Q) I have a table with contain Male and Female, that need to be update if male, then female else male?**

CREATE TABLE gendercheck (gno NUMBER(10), gender char(1));

INSERT INTO gendercheck VALUES (1,'m');

INSERT INTO gendercheck VALUES (2,'f');

INSERT INTO gendercheck VALUES (3,'m');

INSERT INTO gendercheck VALUES (4,'f');

GNO|GENDER|

---+------+

1|m |

2|f |

3|m |

4|f |

SELECT \* FROM gendercheck;

UPDATE gendercheck SET gender=CASE WHEN gender='m' THEN 'F' ELSE 'M' END;

Or

UPDATE gendercheck g1 SET g1.gender=

(SELECT CASE WHEN gender='F' THEN 'm' ELSE 'f' END FROM gendercheck g2

WHERE g2.gno=g1.gno);

Remove duplicates in different ways:

### Use subquery to delete duplicate rows

Here we see an example of using SQL to delete duplicate table rows using an SQL subquery to identify duplicate rows, manually specifying the join columns:

DELETE FROM  
   table\_name A  
WHERE  
  a.rowid >  
   ANY (  
     SELECT  
        B.rowid  
     FROM  
        table\_name B  
     WHERE  
        A.col1 = B.col1  
     AND  
        A.col2 = B.col2  
        );

### Use RANK to delete duplicate rows

This is an example of the RANK function to identify and remove duplicate rows from Oracle tables, which deletes all duplicate rows while leaving the initial instance of the duplicate row:

delete from $table\_name where rowid in

(

select "rowid" from

(select "rowid", rank\_n from

(select rank() over (partition by $primary\_key order by rowid) rank\_n, rowid as "rowid"

from $table\_name

where $primary\_key in

(select $primary\_key from $table\_name

group by $all\_columns

having count(\*) > 1

)

)

)

where rank\_n > 1

)

One of the most important features of Oracle is the ability to detect and remove duplicate rows from a table. While many Oracle DBA place primary key referential integrity constraints on a table, many shops do not use RI because they need the flexibility.

### Use self-join to delete duplicate rows

The most effective way to detect duplicate rows is to join the table against itself as shown below.

select  
   book\_unique\_id,  
   page\_seq\_nbr,  
   image\_key  
from  
   page\_image a  
where  
   rowid >  
     (select min(rowid) from page\_image b  
      where  
         b.key1 = a.key1  
      and  
         b.key2 = a.key2  
      and  
         b.key3 = a.key3  
      );

Please note that you must specify all of the columns that make the row a duplicate in the SQL where clause. Once you have detected the duplicate rows, you may modify the SQL statement to remove the duplicates as shown below:

delete from  
   table\_name a  
where  
   a.rowid >  
   any (select b.rowid  
   from  
      table\_name b  
   where  
      a.col1 = b.col1  
   and  
      a.col2 = b.col2  
   )  
;

### Use analytics to delete duplicate rows

You can also detect and delete duplicate rows using Oracle analytic functions:

delete from  
   customer  
where rowid in  
 (select rowid from  
   (select  
     rowid,  
     row\_number()  
    over  
     (partition by custnbr order by custnbr) dup  
    from customer)  
  where dup > 1);

As we see, there are several ways to detect and delete duplicate rows from Oracle tables

Reader Comments:

### Removing duplicate table rows where rows have NULL values

Rob Arden states:  The tip on this page helped with removing duplicate rows from Oracle tables. I thought this might be useful so I'm passing it on: I needed to add a null check because this fails to remove dupe rows where the fields match on a null value.  So instead of the given:

delete from  
   table\_name a  
where  
   a.rowid >  
   any (select b.rowid  
   from  
      table\_name b  
   where  
      a.col1 = b.col1  
   and  
      a.col2 = b.col2  
   )  
;

I needed to do the following to remove all of the duplicate table rows:

delete from  
   table\_name a  
where  
   a.rowid >  
   any (select b.rowid  
   from  
      table\_name b  
   where  
      (a.col1 = b.col1 or (a.col1 is null and b.col1 is null))  
   and  
      (a.col2 = b.col2 or (a.col2 is null and b.col2 is null))  
   )