Assignment-1

SQL Database Objects Reference Sheet

This reference sheet lists common database objects in SQL along with brief descriptions. These objects define the structure, logic, and security of the database.

|  |  |
| --- | --- |
| Database Object | Description |
| Table | Stores data in rows and columns; the primary storage unit in SQL. |
| View | A virtual table based on the result-set of an SQL query. |
| Index | Improves the speed of data retrieval from a table. |
| Stored Procedure | A saved collection of SQL statements that perform a task. |
| Function | Returns a value and can be used in SQL expressions; similar to procedures. |
| Trigger | Executes automatically in response to certain events on a table or view. |
| Schema | A logical container for database objects; helps organize them. |
| Sequence | Generates a sequence of numeric values, often used for auto-incrementing IDs. |
| Synonym | An alias for another database object to simplify access. |
| Constraint | Rules applied to table columns for data validation and integrity. |

**2**.**what are the metadata tables we have in sql**?

| **Metadata Table** | **Description** |
| --- | --- |
| INFORMATION\_SCHEMA.TABLES | Lists all tables in the database |
| INFORMATION\_SCHEMA.COLUMNS | Lists all columns in all tables |
| INFORMATION\_SCHEMA.SCHEMATA | Lists all schemas |
| INFORMATION\_SCHEMA.CONSTRAINTS | Lists table constraints (PK, FK, UNIQUE, etc.) |
| INFORMATION\_SCHEMA.KEY\_COLUMN\_USAGE | Links constraints to their columns |
| INFORMATION\_SCHEMA.VIEWS | Lists all views in the database |
| INFORMATION\_SCHEMA.ROUTINES | Lists stored procedures and functions |
| INFORMATION\_SCHEMA.TRIGGERS | Lists triggers |

**Vendor-Specific Metadata Tables**

Depending on your database system, there are more detailed system views or tables:

**SQL Server**

| **System View / Table** | **Purpose** |
| --- | --- |
| sys.tables, sys.columns | Lists tables and columns |
| sys.objects, sys.schemas | All database objects and schemas |
| sys.indexes, sys.index\_columns | Index metadata |
| sys.procedures, sys.triggers | Stored procedures and triggers |
| sys.foreign\_keys, sys.check\_constraints | Constraints |

**MySQL**

| **System Table** | **Purpose** |
| --- | --- |
| information\_schema.TABLES | Table metadata |
| information\_schema.STATISTICS | Index metadata |
| mysql.user, mysql.db | User privileges and database access |

**PostgreSQL**

| **System View / Table** | **Purpose** |
| --- | --- |
| pg\_tables, pg\_class | Table information |
| pg\_attribute | Column info |
| pg\_indexes, pg\_index | Index info |
| pg\_constraint | Constraints |
| pg\_proc | Functions and procedures |

**Oracle**

| **System View** | **Purpose** |
| --- | --- |
| ALL\_TABLES, ALL\_TAB\_COLUMNS | Tables and columns |
| USER\_CONSTRAINTS, USER\_INDEXES | Constraints and indexes |
| DBA\_OBJECTS, DBA\_TABLES | All objects in the database |

**SQL Server (T-SQL)**

**1. List All Tables**

sql

CopyEdit

SELECT name, schema\_id

FROM sys.tables;

**2. List All Columns in a Specific Table**

sql

CopyEdit

SELECT name, column\_id, system\_type\_id

FROM sys.columns

WHERE object\_id = OBJECT\_ID('YourTableName');

**3. List Indexes on a Table**

sql

CopyEdit

SELECT i.name AS IndexName, i.type\_desc, c.name AS ColumnName

FROM sys.indexes i

JOIN sys.index\_columns ic ON i.object\_id = ic.object\_id AND i.index\_id = ic.index\_id

JOIN sys.columns c ON ic.object\_id = c.object\_id AND ic.column\_id = c.column\_id

WHERE i.object\_id = OBJECT\_ID('YourTableName');

**4. List All Stored Procedures**

sql

CopyEdit

SELECT name, type\_desc

FROM sys.procedures;

**✅ MySQL**

**1. List All Tables**

sql

CopyEdit

SELECT table\_name

FROM information\_schema.tables

WHERE table\_schema = 'your\_database';

**2. List All Columns in a Table**

sql

CopyEdit

SELECT column\_name, data\_type, is\_nullable

FROM information\_schema.columns

WHERE table\_name = 'your\_table' AND table\_schema = 'your\_database';

**3. List Indexes on a Table**

sql

CopyEdit

SHOW INDEXES FROM your\_table;

**4. List Stored Procedures**

sql

CopyEdit

SELECT routine\_name, routine\_type

FROM information\_schema.routines

WHERE routine\_schema = 'your\_database';

**✅ PostgreSQL**

**1. List All Tables**

sql

CopyEdit

SELECT tablename

FROM pg\_tables

WHERE schemaname NOT IN ('pg\_catalog', 'information\_schema');

**2. List Columns in a Table**

sql

CopyEdit

SELECT column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'your\_table';

**3. List Indexes**

sql

CopyEdit

SELECT indexname, indexdef

FROM pg\_indexes

WHERE tablename = 'your\_table';

**4. List All Functions**

sql

CopyEdit

SELECT proname, proargtypes

FROM pg\_proc

JOIN pg\_namespace ON pg\_proc.pronamespace = pg\_namespace.oid

WHERE nspname NOT IN ('pg\_catalog', 'information\_schema');

**✅ Oracle**

**1. List All Tables**

sql

CopyEdit

SELECT table\_name FROM user\_tables;

**2. List Columns in a Table**

sql

CopyEdit

SELECT column\_name, data\_type

FROM user\_tab\_columns

WHERE table\_name = 'YOUR\_TABLE';

**3. List Constraints**

sql

CopyEdit

SELECT constraint\_name, constraint\_type, table\_name

FROM user\_constraints;

**4. List Indexes**

sql

CopyEdit

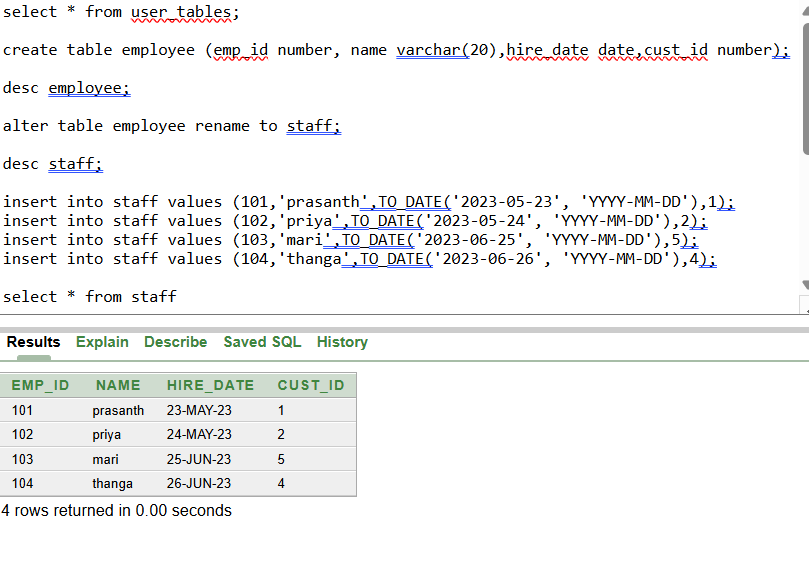
SELECT index\_name, table\_name

FROM user\_indexes;

**Assignment-2**

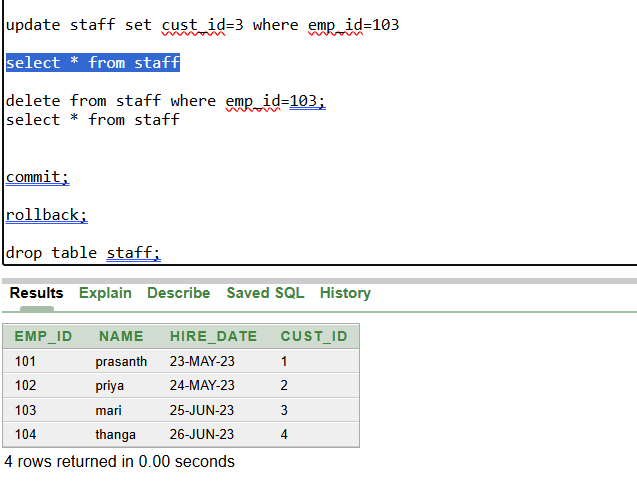
**1.create table and change table name ,update,commit,delete,rollback concepts?**

**After created and alter inserted in table:**

****

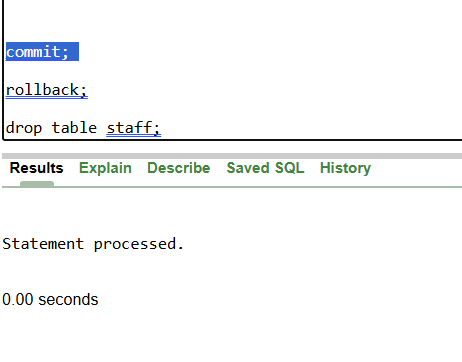
INSERTED VALUES IN TABLE

**Updated cust\_id=3 value in staff table:**

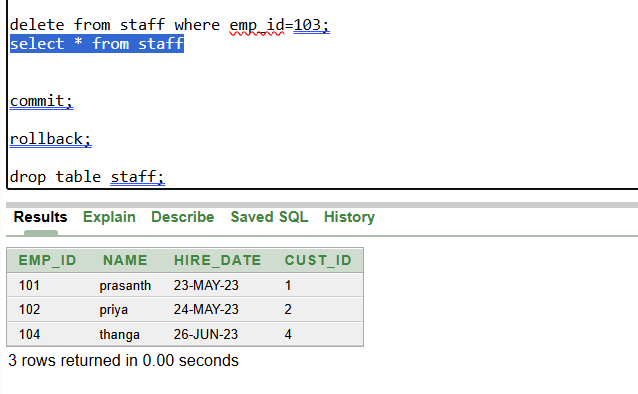
****

UPDATED VALUES

**Commit the table after updating:**

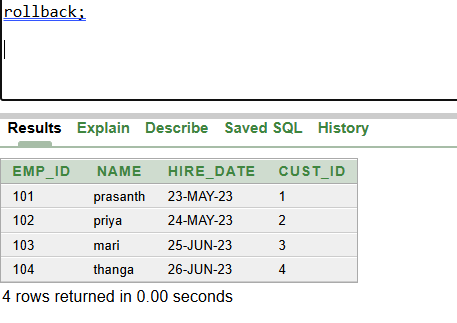
****

**Delete the emp\_id =103 in table:**

****

DELETED 103 ROW

**Rollback is to retrieve deleted data from table but before there shouldn’t be commited otherwise rollback wont comes in play…**

****

RESTORED EMP\_ID=103;

**2.create table and checking how its works:**

**create table student(id number,name varchar(20),subject varchar(20),marks number); (table created comes under DDL so AUTOCOMMIT )**

**insert into student(id, name,subject,marks) values (1001,'jyo','maths',80)(table insert comes under DML so no autocommit)**

**insert into student(id, name,subject,marks) values (1002,'pras','maths',85) (table insert comes under DML so no autocommit)**

**insert into student(id, name,subject,marks) values (1003,'master','maths',90) (table insert comes under DML so no autocommit)**

**select \* from student;**

**create table student1(id number,name varchar(20),subject varchar(20),marks number); (table created comes under DDL so AUTOCOMMIT )**

**insert into student1(id, name,subject,marks) values (1001,'jyo','maths',80) (table insert comes under DML so no autocommit)**

**insert into student1(id, name,subject,marks) values (1002,'pras','maths',85) (table insert comes under DML so no autocommit)**

**truncate table student1;(TRUNCATED ONLY MENTIONED TABLE STUDENT1)**

**select \* from student1;**

**rollback;(HERE TABLE1 INSERTION IS NOT COMMIT SO IT WILL ROLLBACK TO EMPTY FOR STUDENT TABLE AND ANOTHER TABLE ALREADY TRUNCATED….)**

**create table student2(id number,name varchar(20),subject varchar(20),marks number); (table created comes under DDL so AUTOCOMMIT )**

**insert into student2(id, name,subject,marks) values (1001,'jyo','maths',80) (table insert comes under DML so no autocommit)**

**insert into student2(id, name,subject,marks) values (1002,'pras','maths',85) (table insert comes under DML so no autocommit)**

**commit; (HERE STUDENT2 TABLE IS COMMIT)**

**select \* from student2;**

**SO, FINALLY**

**STUDENT TABLE ROLLBACK**

**STUDENT1 TABLE TRUNCATED**

**STUDENT2 TABLE COMMIT**

**Assignment-3**

**1.check wheather we can use multiple constraints in single column?**

**Yes, you can apply multiple constraints to a single column**

**Ex;**

**CREATE TABLE employees (**

**emp\_id INT PRIMARY KEY,**

**email VARCHAR(100) NOT NULL UNIQUE,**

**salary DECIMAL(10, 2) NOT NULL CHECK (salary > 0)**

**);**

**Here Common column-level constraints you can combine:**

**Not null**

**Unique**

**Default**

**Check**

**But here,primary key should be in for one field only and that should associate with reference foreign key if needed**

**So, we cant use multiple primary key in single column,, remaining keys we can use multiple times**

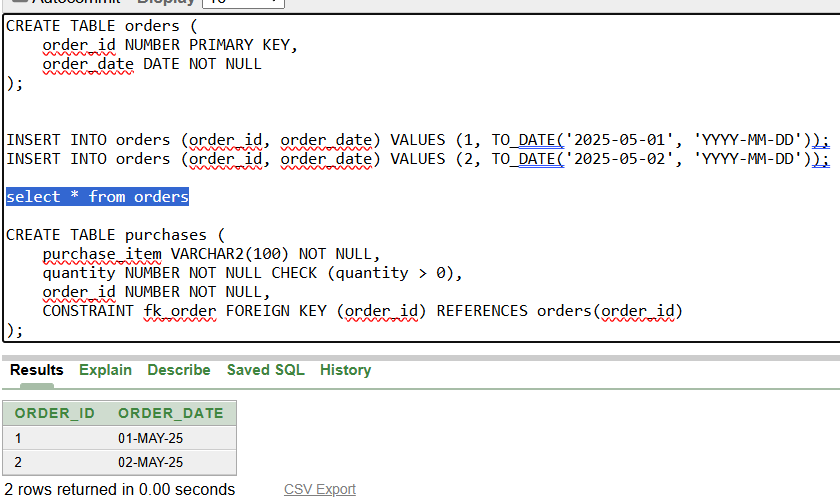
**Composite Primary Key: we have this key to use one or more column to identify unique values….**

**PRIMARY KEY (student\_id, course\_id)**

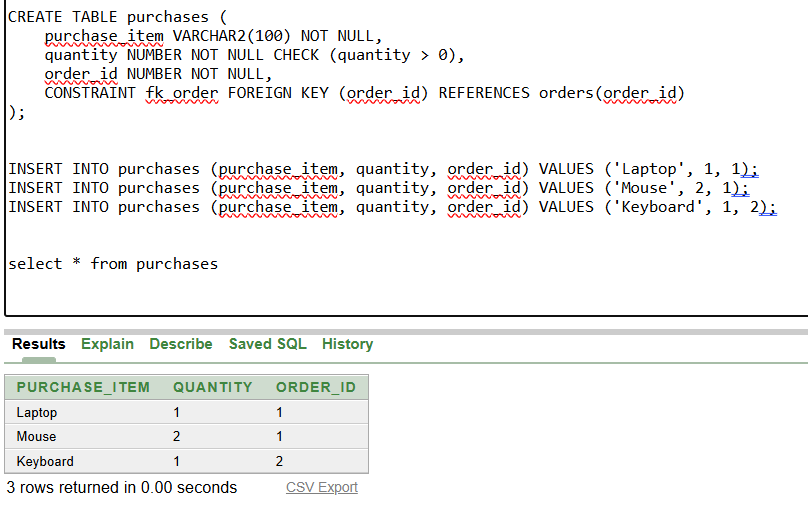
**Here, student\_id cant able to find unique value so we associate with another course\_id to filter perfect unique value**

**2.create two table using constraints associated?**

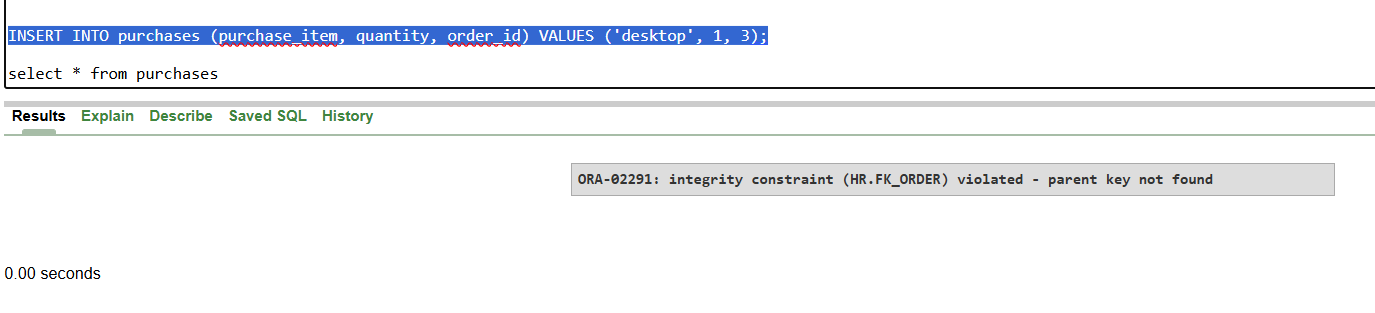
**Create order table and insert value:**

****

**Create purchase table and insert values:**

****

**Check whether primary and foreign key associated correctly**

****

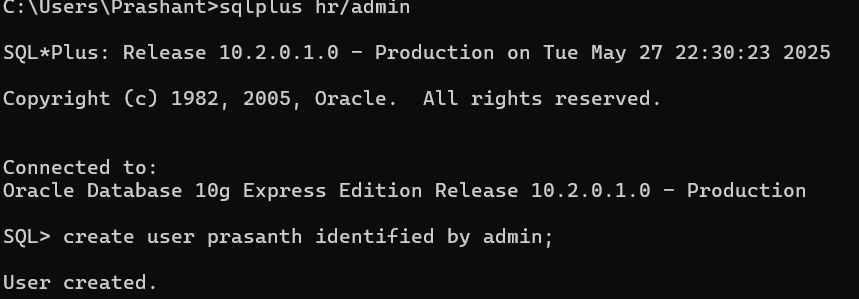
**Here, I change purchase order\_id to 3 which is not in primary key orders table errors:parent key not found**

**So,here confirmed foreign key is reference to primary key order table and if we want to use that means their should be common id with unique value association is must**

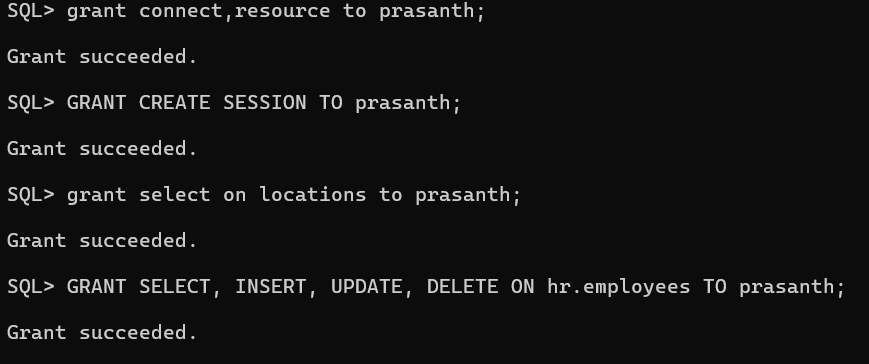
**Assignment-4**

**1.create user access from hr to user, user to user1 and access hr tables from user1?**

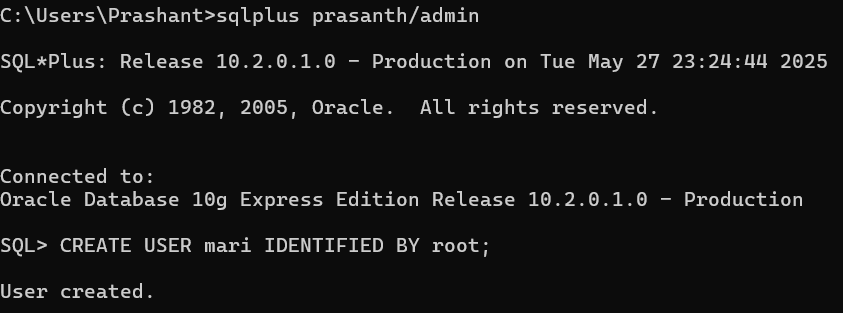
**Create hr to user(prasanth):**

****

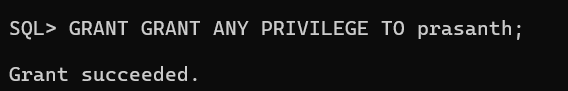
**Grand access:**

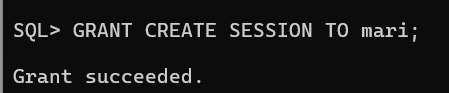
****

**create user1(mari) from user(prasanth)**

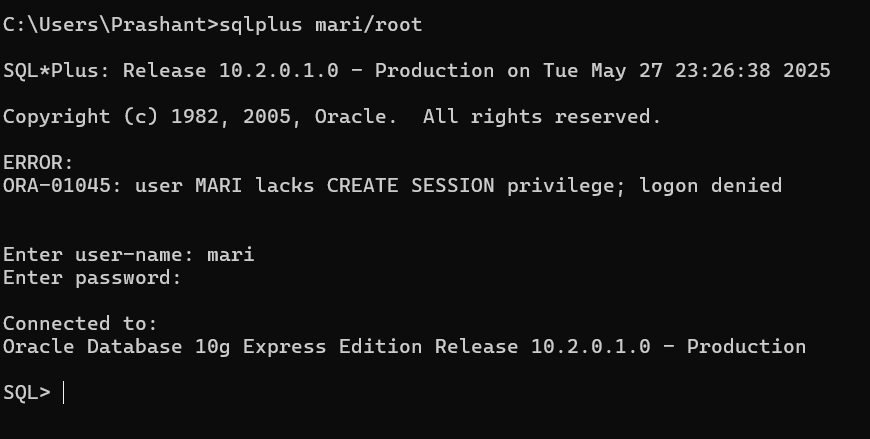
****

**But to start session for user1(mari) I used PRIVILEGE option to user(prasanth):**

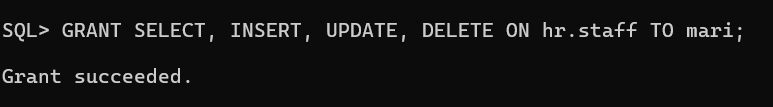
****

****

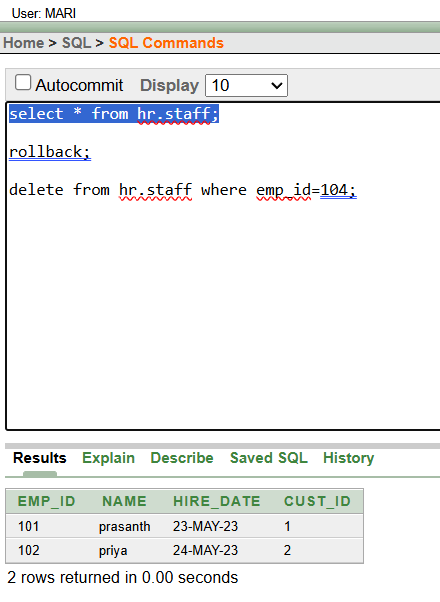
**Now from user(prasanth) to user1(mari) successfully connected:**

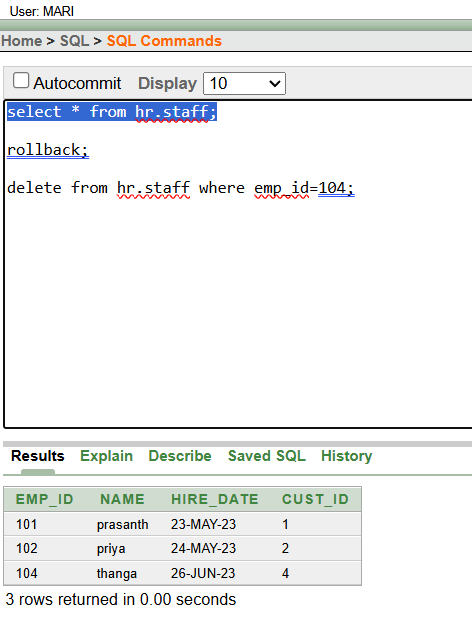
****

**Now task is to access hr table to user1(mari):**

****

**Here I deleted and rollback data of hr.staff table in user1(mari):**

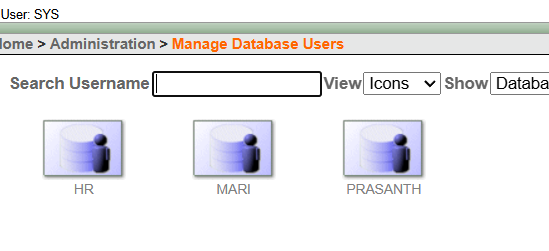
****

****

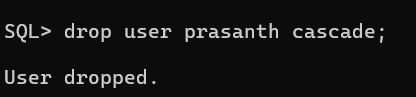
**So,successfully accessed hr table using user1(mari)**

**2.how to drop or delete user in database?**

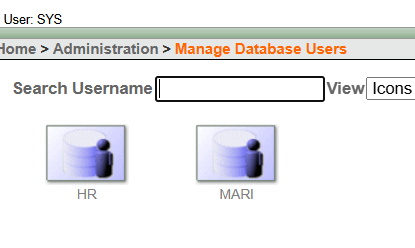
**Here 3 users I have :**

****

**After disconnecting user(prasanth):**

****

**After droping user(prasanth) in manage users we have only:**

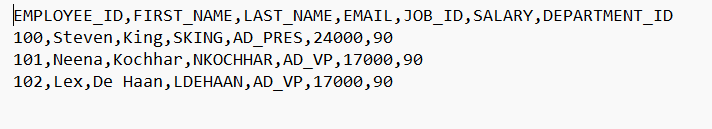
****

**So,successfully droped user(prasanth)…**

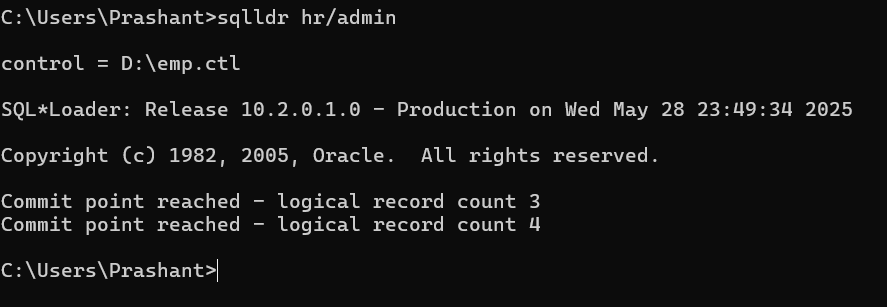
**Assignment-5**

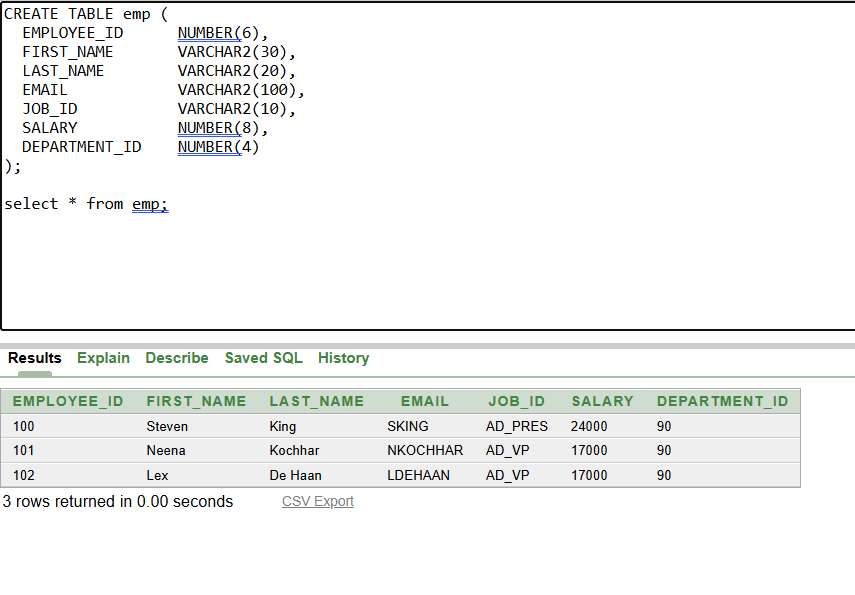
**1.load 3 datas in table and again load 3 datas in same table?**

**Loading csv(comma separate value file) using ctl(control file) thru command line:**

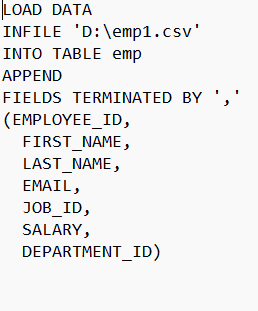
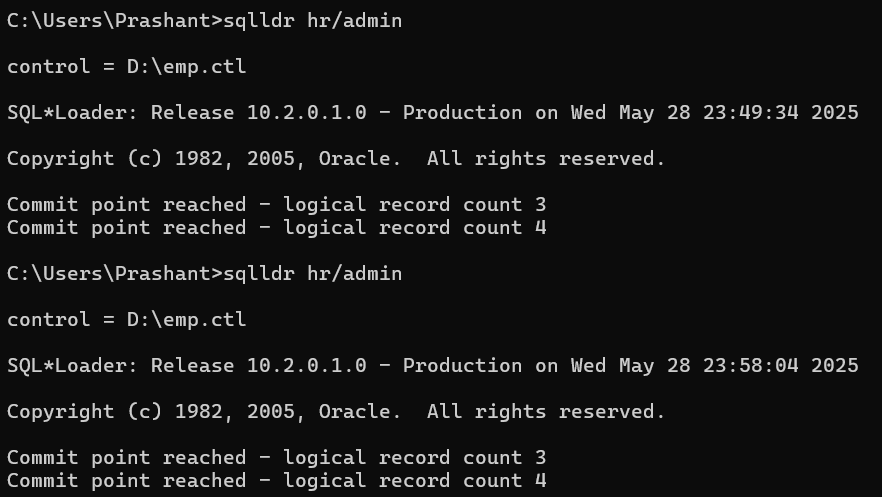
****

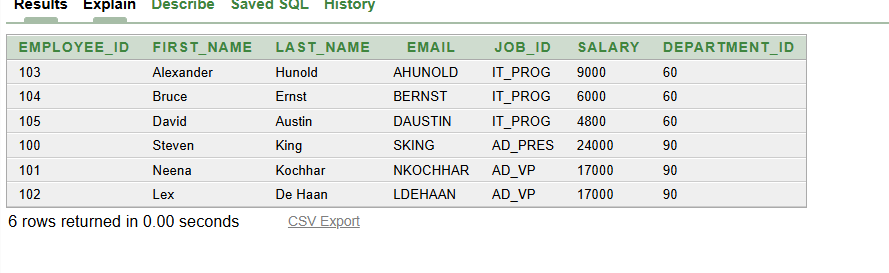
**CSV FILE**

****

****

**Again adding using append command:**

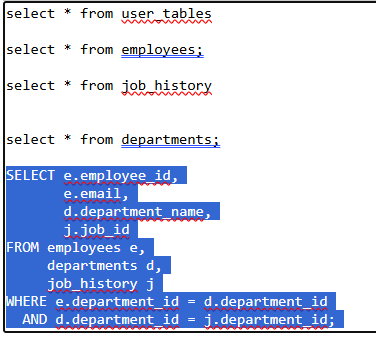
****

****

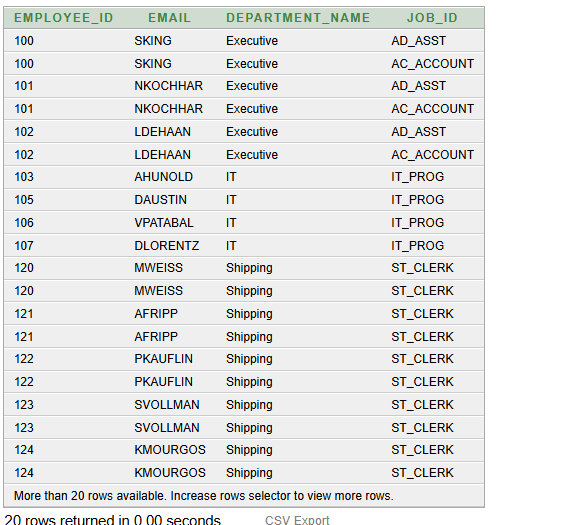
**Assignment-6**

**1.do joins for 3 tables only matching tables to view:**

**Here employees,departments and job\_history has common department\_id to join:**

****

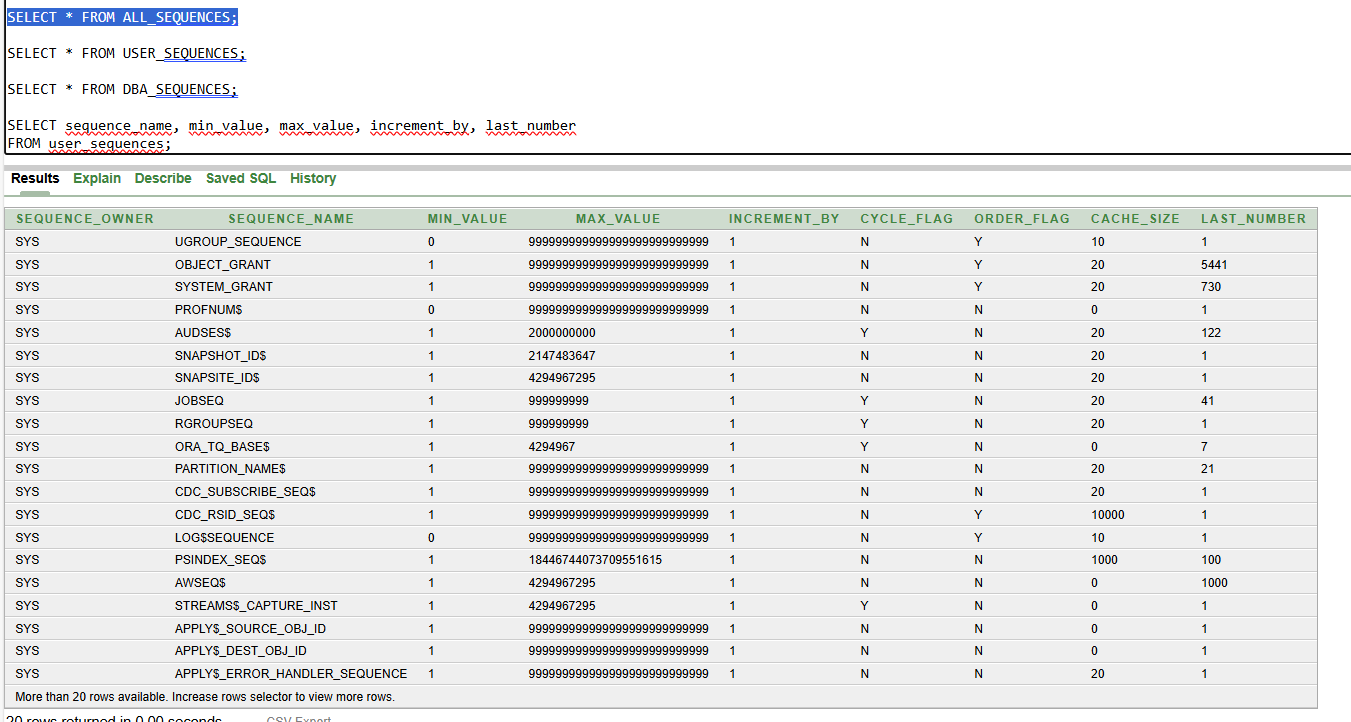
**OUTPUT:**

****

**Assignment-7**

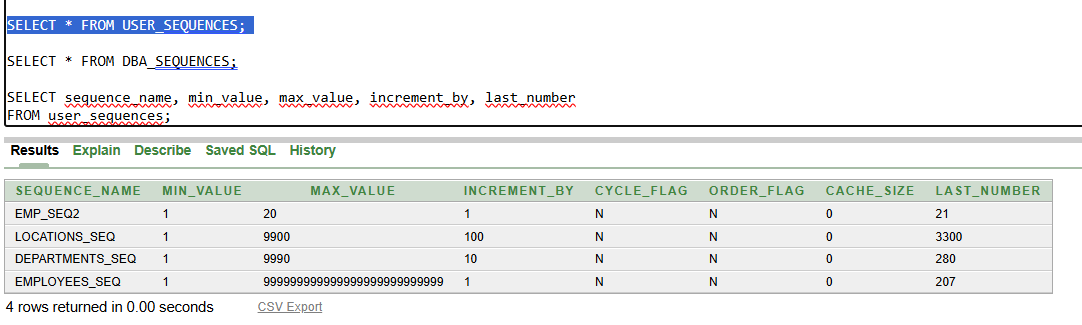
1. **find metadata for sequence to know what are the sequence available in current user?**

**View all sequences accessible to the current user: SELECT \* FROM ALL\_SEQUENCES;**

****

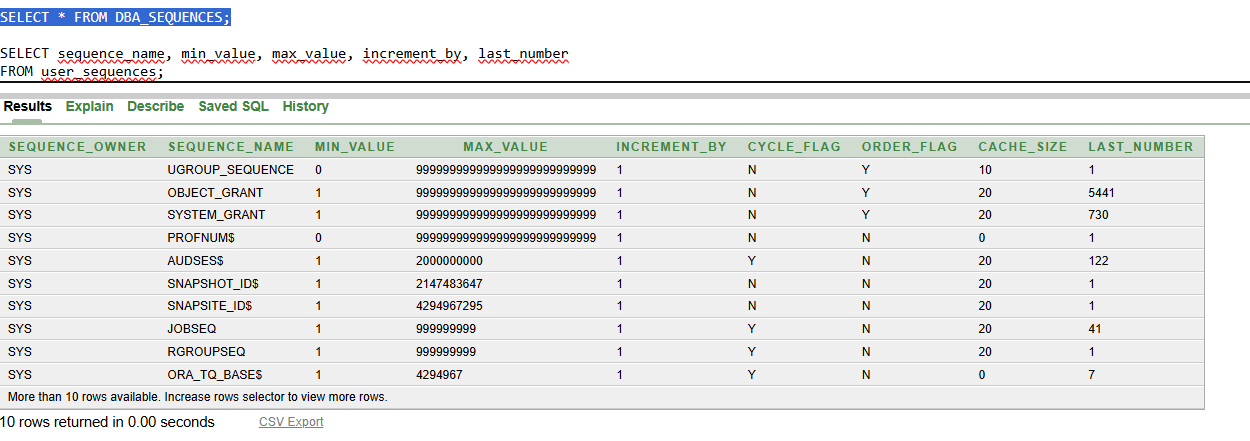
**This view includes all sequences that the current user has access to, including those owned by other users**

**View only the sequences owned by the current user:** **SELECT \* FROM USER\_SEQUENCES;**

****

**This view is filtered to only show sequences owned by the current user.**

**View all sequences in the database same as all sequences:** **SELECT \* FROM DBA\_SEQUENCES;**

****

**SEQUENCE\_NAME: Name of the sequence.**

**MIN\_VALUE: Minimum value of the sequence.**

**MAX\_VALUE: Maximum value.**

**INCREMENT\_BY: The increment step.**

**CYCLE\_FLAG (Y/N):**

* **'Y' means the sequence restarts from the MINVALUE after reaching MAXVALUE.**
* **'N' means it will throw an error once it exceeds MAXVALUE.**

**ORDER\_FLAG (Y/N):**

* **'Y' guarantees that sequence numbers are generated in order, important in RAC (Real Application Clusters) environments.**
* **'N' does not guarantee strict ordering (but is faster).**

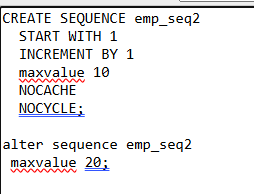
**CACHE\_SIZE:**

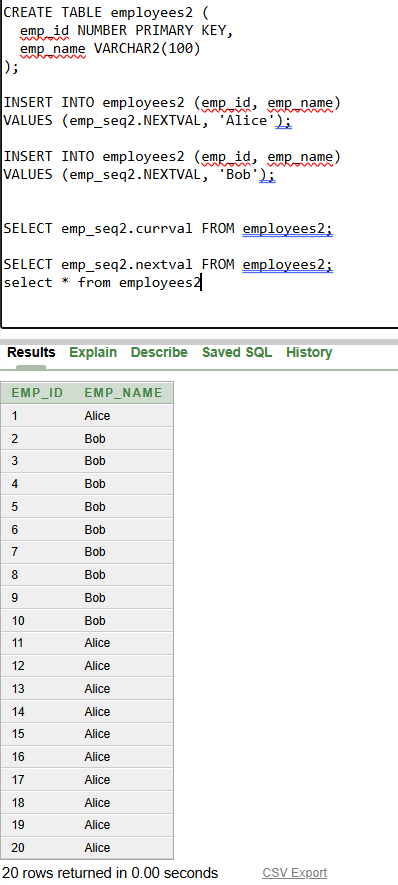
* **Number of sequence numbers kept in memory for faster access. Higher cache reduces disk access but risks value loss in crash.**

**LAST\_NUMBER:**

* **Indicates the next number to be generated after the cached values are exhausted. It may not reflect the most recently generated number if caching is in use.**

1. **create a table and insert value by using sequence?**

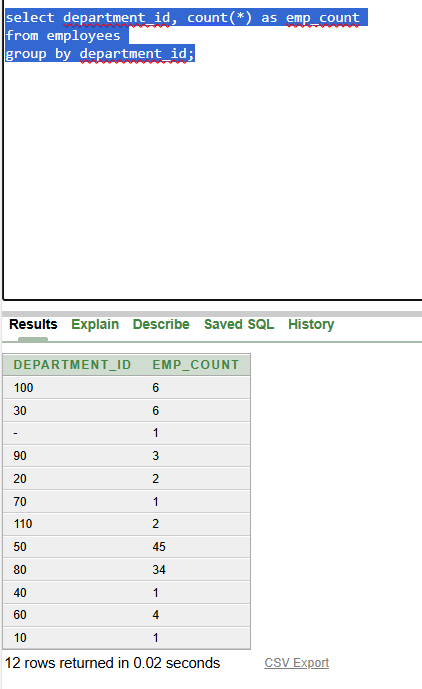
****

****

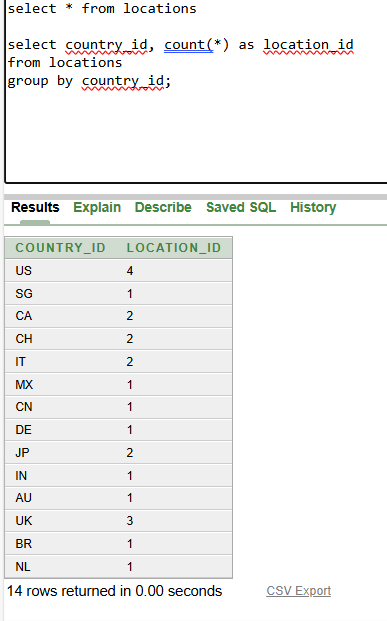
**Assignment-8**

**GROUP BY**

1. **HOW MANY EMPLOYEE WORKING DEPARTMENT WISE?**

****

**based on country\_id in location table:**

****

**Assignment-9**

**To find what are the views available in schema, what are the metadata table we using:**

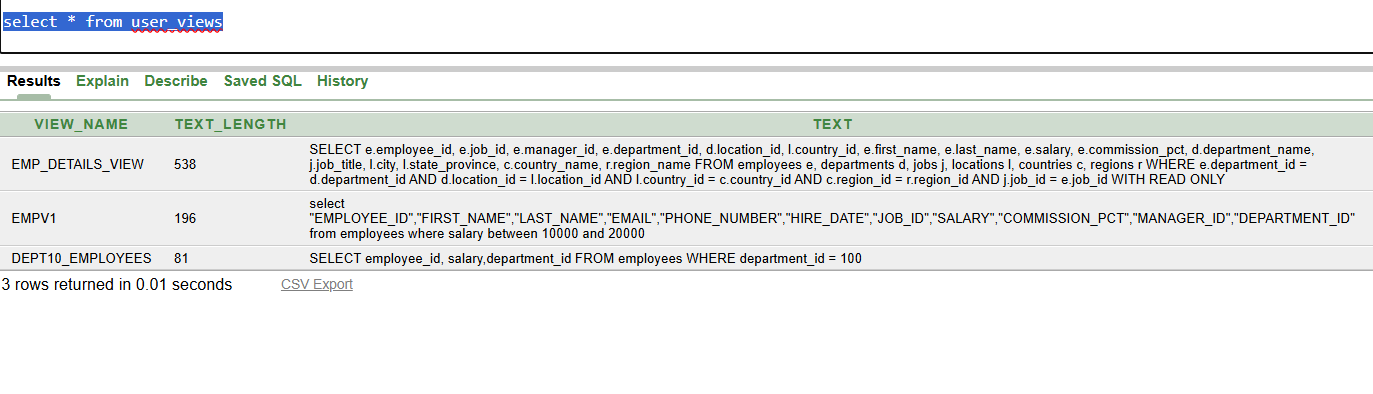
**To find all views available in a schema in Oracle Database,**

**you can query the data dictionary views (metadata**

**tables). Here are the main ones used:**

**1. USER\_VIEWS:**

**Lists all views owned by the current user.**

****

**2. ALL\_VIEWS:**

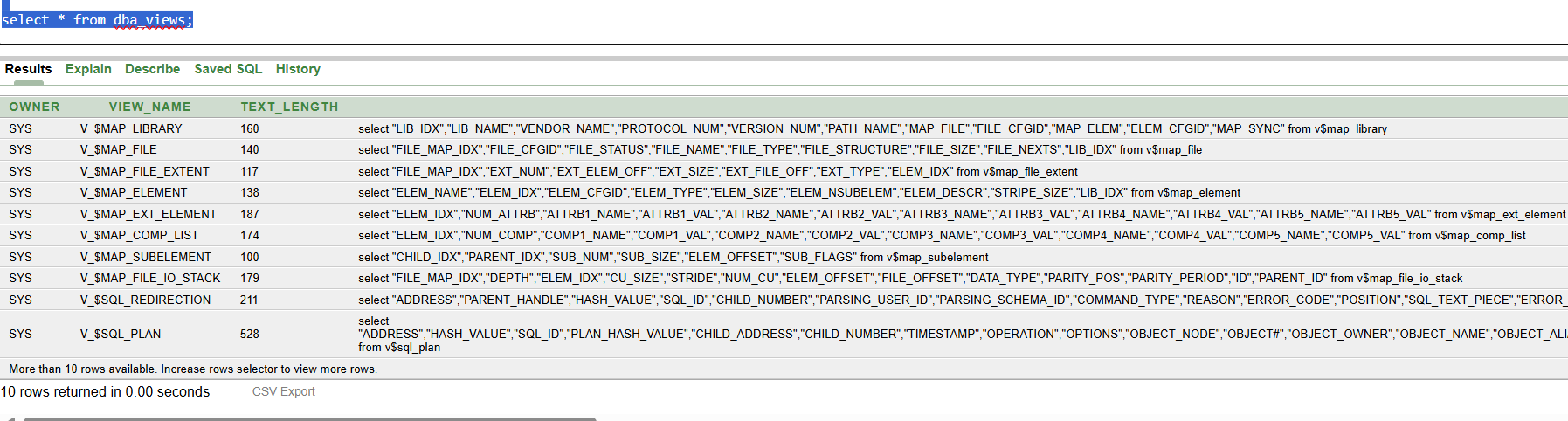
**Lists all views accessible to the current user (including views owned by others if you have privileges).**

****

**3. DBA\_VIEWS:**

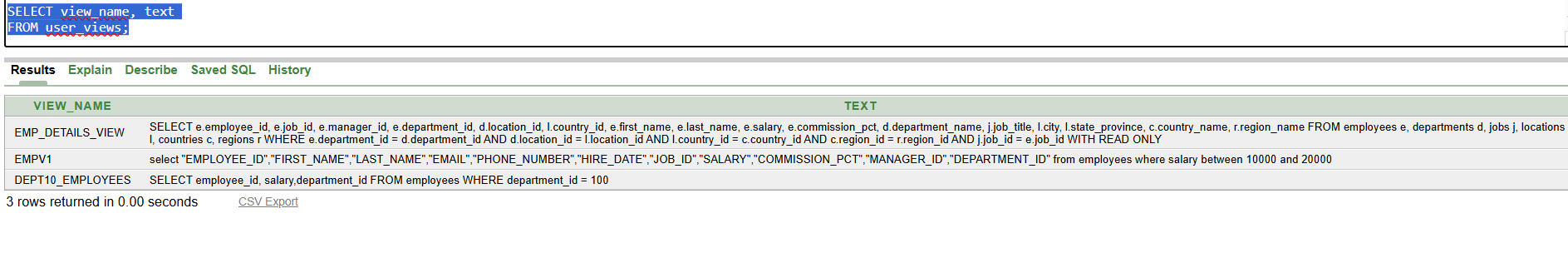
**Lists all views in the entire database, across all**

**schemas. *(requires DBA privileges)***

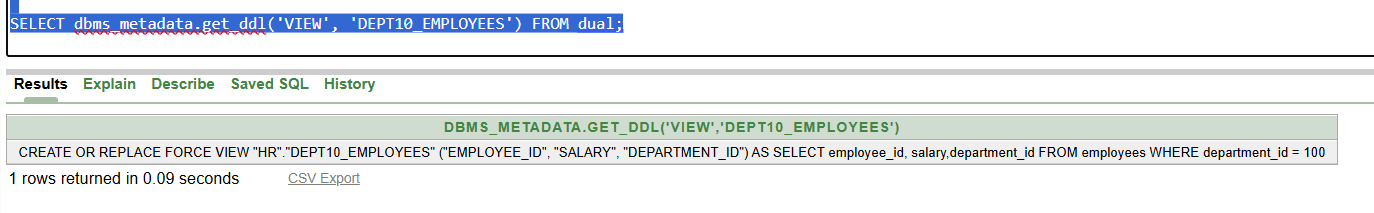
****

| **Column Name** | **Description** |
| --- | --- |
| **VIEW\_NAME** | **Name of the view** |
| **TEXT** | **SQL query text that defines the view** |
| **OWNER** | **Owner of the view (only in ALL\_ / DBA\_)** |

**Example: Show All Views and Their SQL Definition:**

****

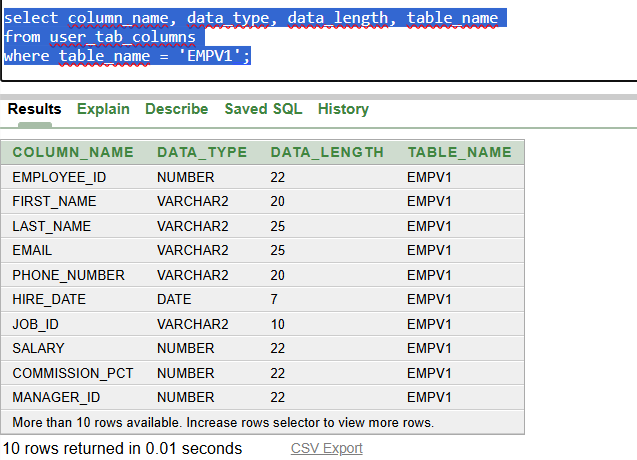
**To see the SQL query behind a view in a formatted way, use:**

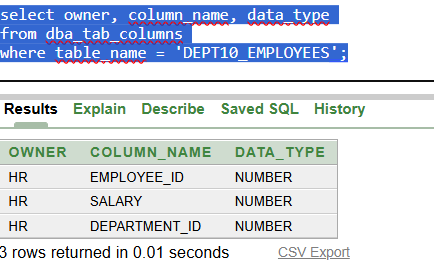
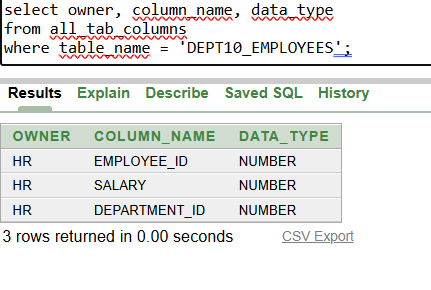
****

**4. Get the Columns of a View**

**Use the USER\_TAB\_COLUMNS, ALL\_TAB\_COLUMNS, or**

**DBA\_TAB\_COLUMNS metadata views:**

****

****

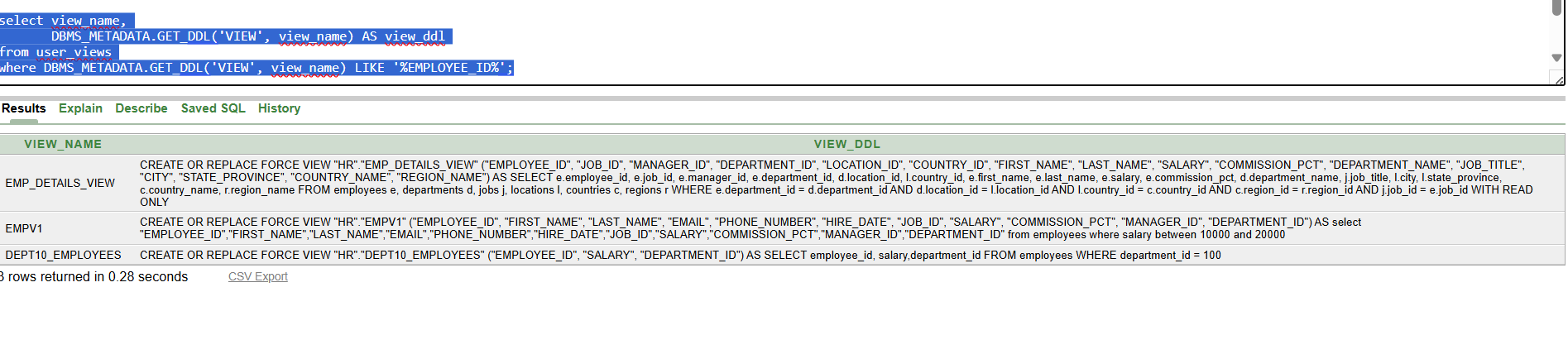
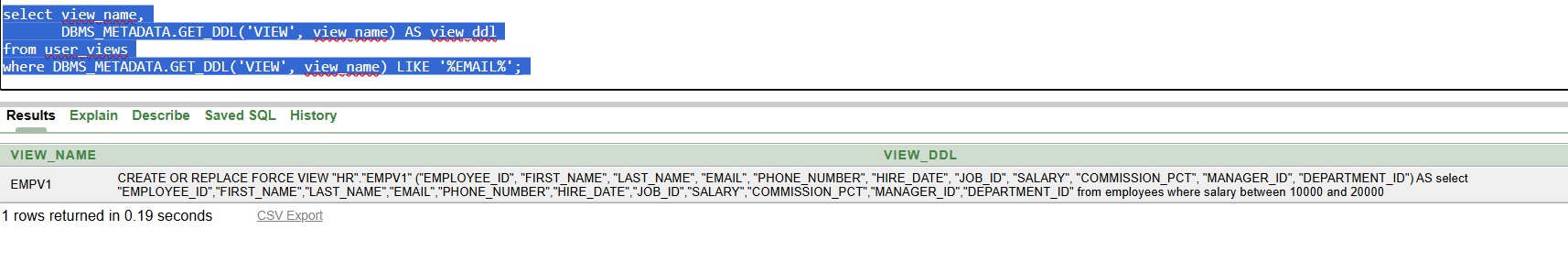
**View names are stored in uppercase by default in Oracle.**

**5.** **Find Views That Use a Specific Table:**

**Example: Find views that use the EMPLOYEES table:**

**DBMS\_METADATA.GET\_DDL works for all version to search specific table VIEWS**

**Here I used for EMPLOYEE\_ID and EMAIL:**

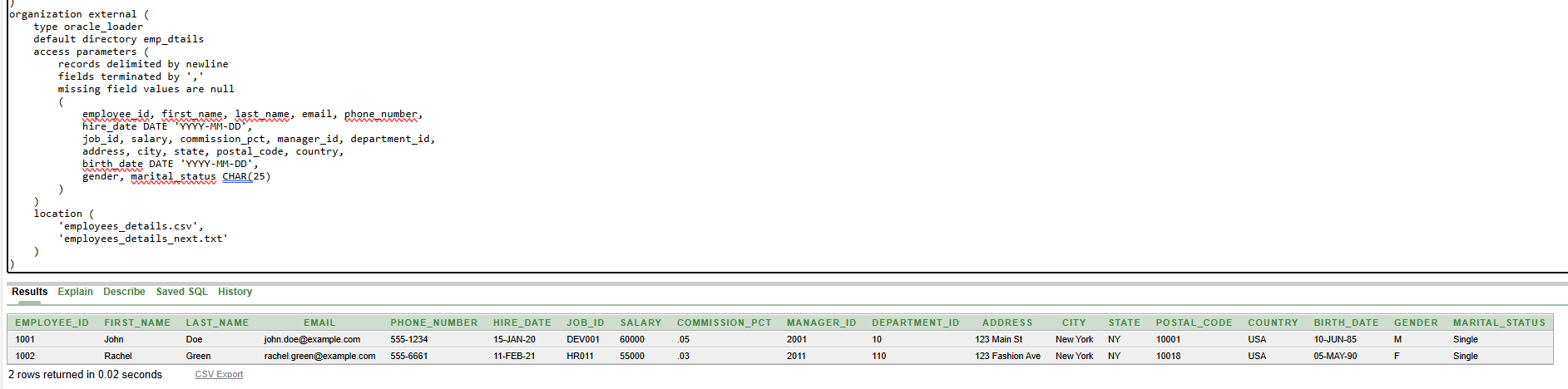
****

**Summary of Metadata Tables Used for Views**

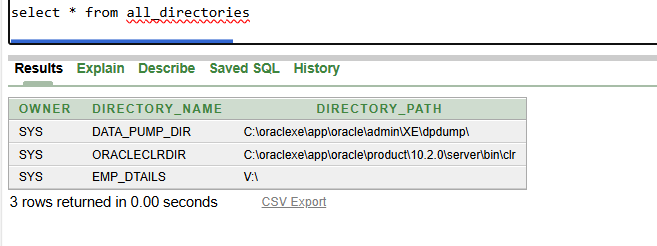
| **Purpose** | **Metadata View** |
| --- | --- |
| **List views in current schema** | **USER\_VIEWS** |
| **List all views accessible to user** | **ALL\_VIEWS** |
| **List all views in the database** | **DBA\_VIEWS** |
| **Find view definitions (SQL)** | **TEXT column or DBMS\_METADATA.GET\_DDL** |
| **List columns in a view** | **USER\_TAB\_COLUMNS** |
| **List view columns across all users** | **ALL\_TAB\_COLUMNS / DBA\_TAB\_COLUMNS** |

**Assignment-10**

**How to add external tables using csv and txt file:**

****

**Selecting all directories:**

****

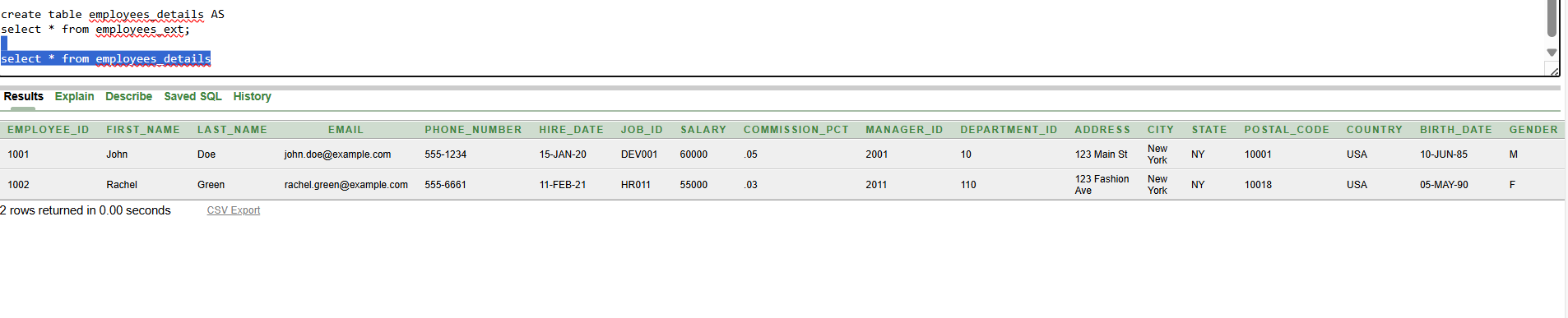
**When working on external tables the most commonly used format is CSV**

**Droping directories is possible only,when we have DROP ANY DIRECTORY privilege access:**

**drop directory directory\_name;**

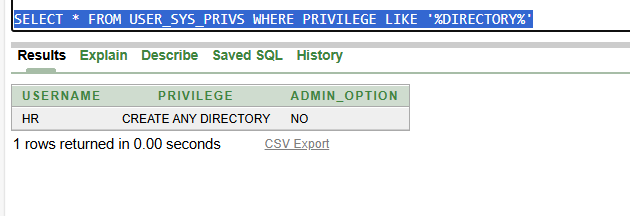
**still,** **this command only removes the reference in Oracle, not the actual folder on the filesystem.**

**If you want to move data from external to permanent table:**

****

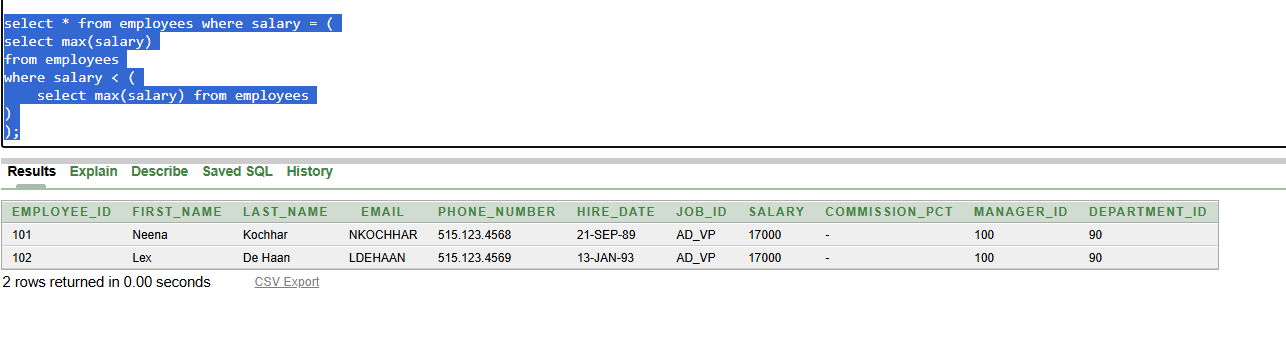
**Now employees\_details is a normal Oracle table.**

**To check current users privileges:**

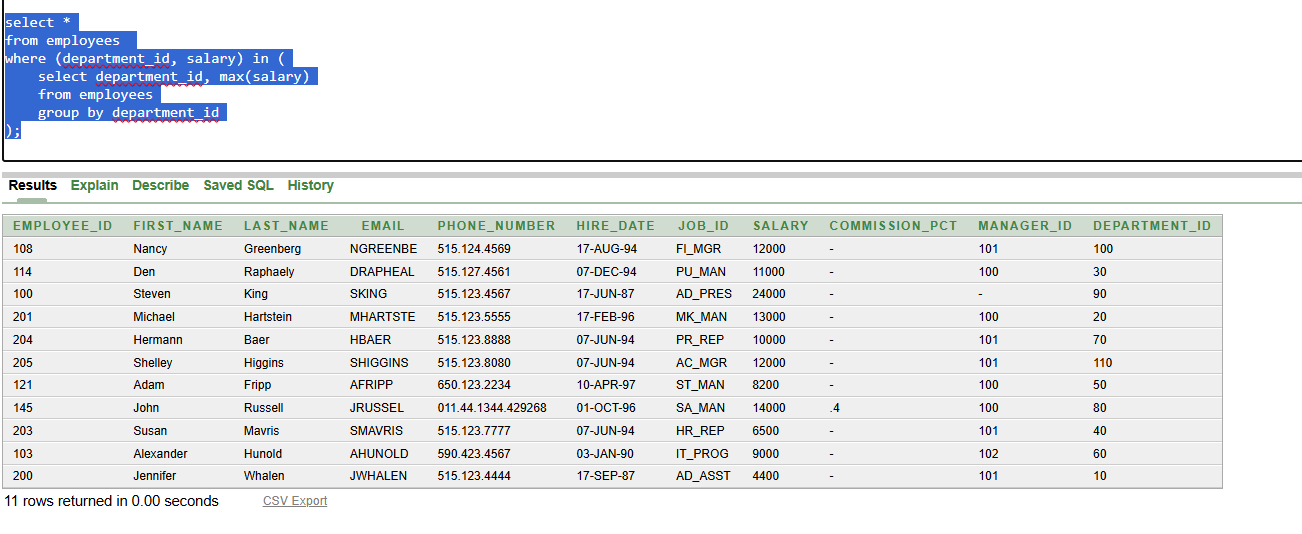
****

**Assignment-11**

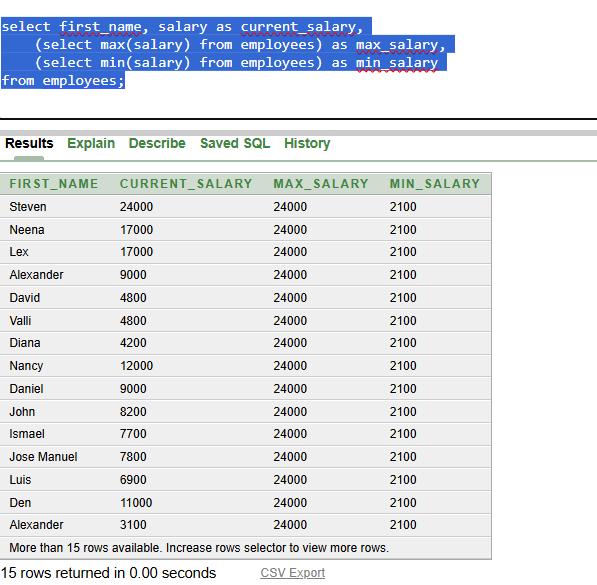
**1.find the 2nd highest salary using subquery and display all details?**

****

**2.** **who are all get highest salary from department wise using multi column subquery?**

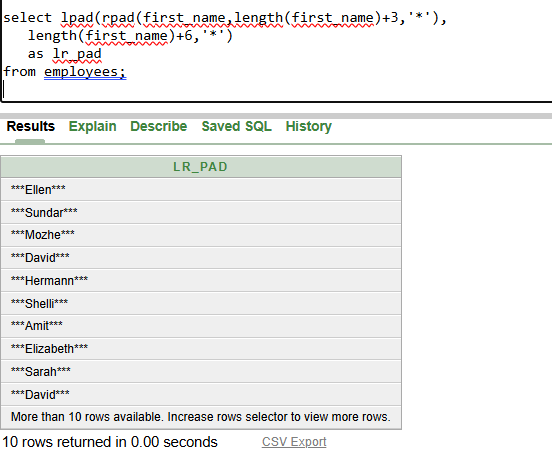
****

**3.** **current salary of employees, max\_salary and min\_salary of employees and display first\_name,salary,max\_salary,min\_salary?**

****

**Assignment-12**

**1.need to print \*\*\* in before and after string using lpad &rpad**

****

**lpad,rpad(string, total\_length,pad\_string)**

**Explaination: for ellen**

**Inner fn execute first rpad(ellen,5+3=8,’\*’) ans = ellen\*\*\***

**Then in lpad(ellen\*\*\* as a string, 5+6=11,’\*’)=\*\*\*ellen\*\*\***

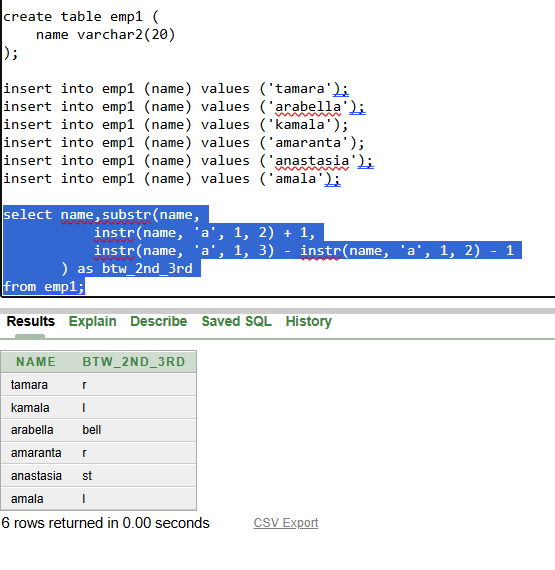
**here updated rpad string length became 8 so we adding +6 in actual string to tally rpad additional +3**

**output:**

**\*\*\*ellen\*\*\***

**2.** **create table in the name of emp add a column in name of**

**NAME insert 3values print words in between 2 and 3 (A)?**

****

**Explaination:**

**For tamara first we execute inside substr to find first position and length of the string so,**

**Syntax of substr:**

**Substr(string, first\_position, length)s**

**Finding First\_position:**

**we use instr,syntax**

**Instr(string, substring, start\_pos, occurance):**

**Instr(tamara, ‘a’, 1 , 2) 🡺pos = 4 but we need next value of a so +1**

**First\_position = Instr(tamara, ‘a’, 1 , 2) +1 = 4+1 = 5**

**Length:**

**Logic: (3rd pos - 2ndpos - 1) 🡺to find length of the string in between**

**instr(tamara, ‘a’, 1, 3) - instr(tamara, ‘a’, 1, 2) - 1 = 1**

**6 - 4 - 1 = 1**

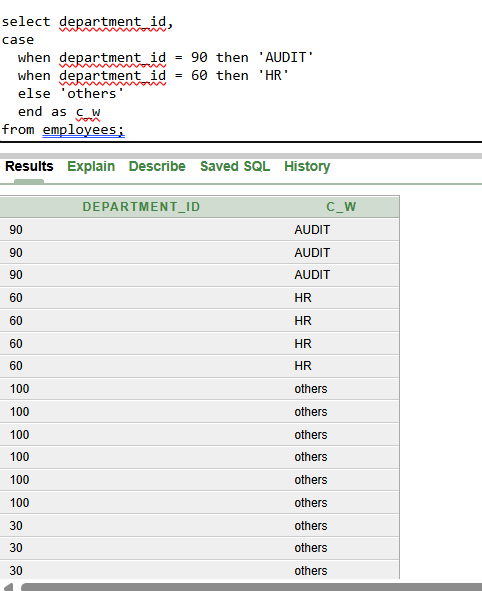
**So finally we got substr(tamara, 5,1)**

**Output:**

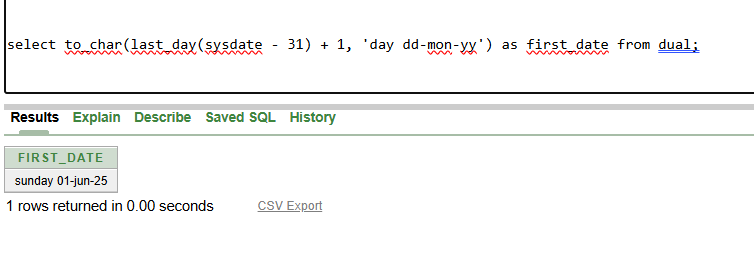
**r**

**assignment-13**

1. **Try to print others by using case statement:**

****

1. **print the first\_date of the month**

****