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

Metadata Table	Description
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

System View / Table	Purpose
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

System View	Purpose
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:

```
select * from user_tables;

create table employee (emp_id number, name varchar(20),hire_date date,cust_id number);

desc employee;

alter table employee rename to staff;

desc staff;

insert into staff values (101,'prasanth',TO_DATE('2023-05-23', 'YYYY-MM-DD'),1);
insert into staff values (102,'priya',TO_DATE('2023-05-24', 'YYYY-MM-DD'),2);
insert into staff values (103,'mari',TO_DATE('2023-06-25', 'YYYY-MM-DD'),5);
insert into staff values (104,'thanga',TO_DATE('2023-06-26', 'YYYY-MM-DD'),4);

select * from staff
```

Results | Explain | Describe | Saved SQL | History

EMP_ID	NAME	HIRE_DATE	CUST_ID
101	prasanth	23-MAY-23	1
102	priya	24-MAY-23	2
103	mari	25-JUN-23	5
104	thanga	26-JUN-23	4

4 rows returned in 0.00 seconds

INSERTED VALUES IN TABLE

Updated cust_id=3 value in staff table:

```
update staff set cust_id=3 where emp_id=103

select * from staff

delete from staff where emp_id=103;
select * from staff


commit;

rollback;

drop table staff;
```

Results Explain Describe Saved SQL History

EMP_ID	NAME	HIRE_DATE	CUST_ID
101	prasanth	23-MAY-23	1
102	priya	24-MAY-23	2
103	mari	25-JUN-23	3
104	thanga	26-JUN-23	4

UPDATED VALUES

4 rows returned in 0.00 seconds

Commit the table after updating:

```
commit;

rollback;

drop table staff;
```

Results Explain Describe Saved SQL History

Statement processed.

0.00 seconds

Delete the emp_id =103 in table:

```
delete from staff where emp_id=103;
select * from staff


commit;

rollback;

drop table staff;
```

Results Explain Describe Saved SQL History

EMP_ID	NAME	HIRE_DATE	CUST_ID
101	prasanth	23-MAY-23	1
102	priya	24-MAY-23	2
104	thanga	26-JUN-23	4

3 rows returned in 0.00 seconds

DELETED 103 ROW

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

```
rollback;
```

Results | Explain | Describe | Saved SQL | History

EMP_ID	NAME	HIRE_DATE	CUST_ID
101	prasanth	23-MAY-23	1
102	priya	24-MAY-23	2
103	mari	25-JUN-23	3
104	thanga	26-JUN-23	4

RESTORED EMP_ID=103;

4 rows returned in 0.00 seconds

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 TABLE orders (
    order_id NUMBER PRIMARY KEY,
    order_date DATE NOT NULL
);


INSERT INTO orders (order_id, order_date) VALUES (1, TO_DATE('2025-05-01', 'YYYY-MM-DD'));
INSERT INTO orders (order_id, order_date) VALUES (2, TO_DATE('2025-05-02', 'YYYY-MM-DD'));

select * from orders

CREATE TABLE purchases (
    purchase_item VARCHAR2(100) NOT NULL,
    quantity NUMBER NOT NULL CHECK (quantity > 0),
    order_id NUMBER NOT NULL,
    CONSTRAINT fk_order FOREIGN KEY (order_id) REFERENCES orders(order_id)
);
```

Results Explain Describe Saved SQL History

ORDER_ID	ORDER_DATE
1	01-MAY-25
2	02-MAY-25

2 rows returned in 0.00 seconds CSV Export

Create purchase table and insert values:

```
CREATE TABLE purchases (
    purchase_item VARCHAR2(100) NOT NULL,
    quantity NUMBER NOT NULL CHECK (quantity > 0),
    order_id NUMBER NOT NULL,
    CONSTRAINT fk_order FOREIGN KEY (order_id) REFERENCES orders(order_id)
);


INSERT INTO purchases (purchase_item, quantity, order_id) VALUES ('Laptop', 1, 1);
INSERT INTO purchases (purchase_item, quantity, order_id) VALUES ('Mouse', 2, 1);
INSERT INTO purchases (purchase_item, quantity, order_id) VALUES ('Keyboard', 1, 2);


select * from purchases
```

Results Explain Describe Saved SQL History

PURCHASE_ITEM	QUANTITY	ORDER_ID
Laptop	1	1
Mouse	2	1
Keyboard	1	2

3 rows returned in 0.00 seconds CSV Export

Check whether primary and foreign key associated correctly

```
INSERT INTO purchases (purchase_item, quantity, order_id) VALUES ('desktop', 1, 3);
select * from purchases
```

Results Explain Describe Saved SQL History

```
ORA-02291: integrity constraint (HR.FK_ORDER) violated - parent key not found
```

0.00 seconds

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):

```
C:\Users\Prashant>sqlplus hr/admin

SQL*Plus: Release 10.2.0.1.0 - Production on Tue May 27 22:30:23 2025

Copyright (c) 1982, 2005, Oracle.  All rights reserved.


Connected to:
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production

SQL> create user prasanth identified by admin;

User created.
```

Grand access:

```
SQL> grant connect,resource to prasanth;

Grant succeeded.

SQL> GRANT CREATE SESSION TO prasanth;

Grant succeeded.

SQL> grant select on locations to prasanth;

Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON hr.employees TO prasanth;

Grant succeeded.
```

create user1(mari) from user(prasanth)

```
C:\Users\Prashant>sqlplus prasanth/admin

SQL*Plus: Release 10.2.0.1.0 - Production on Tue May 27 23:24:44 2025

Copyright (c) 1982, 2005, Oracle.  All rights reserved.


Connected to:
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production

SQL> CREATE USER mari IDENTIFIED BY root;

User created.
```

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

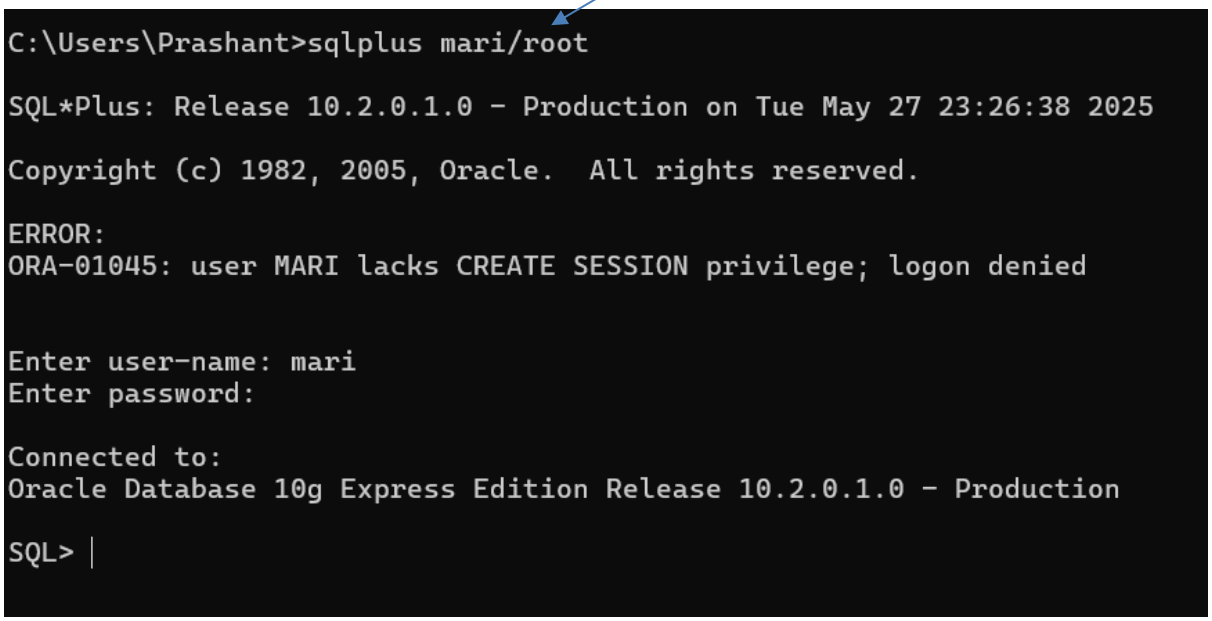
```
SQL> GRANT GRANT ANY PRIVILEGE TO prasanth;

Grant succeeded.
```

```
SQL> GRANT CREATE SESSION TO mari;

Grant succeeded.
```

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



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

```
SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON hr.staff TO mari;

Grant succeeded.
```

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

User: MARI

Home > SQL > SQL Commands

Autocommit Display 10

```
select * from hr.staff;

rollback;

delete from hr.staff where emp_id=104;
```

Results Explain Describe Saved SQL History

EMP_ID	NAME	HIRE_DATE	CUST_ID
101	prasanth	23-MAY-23	1
102	priya	24-MAY-23	2

2 rows returned in 0.00 seconds

User: MARI

Home > SQL > SQL Commands

Autocommit Display 10

```
select * from hr.staff;

rollback;

delete from hr.staff where emp_id=104;
```

Results Explain Describe Saved SQL History

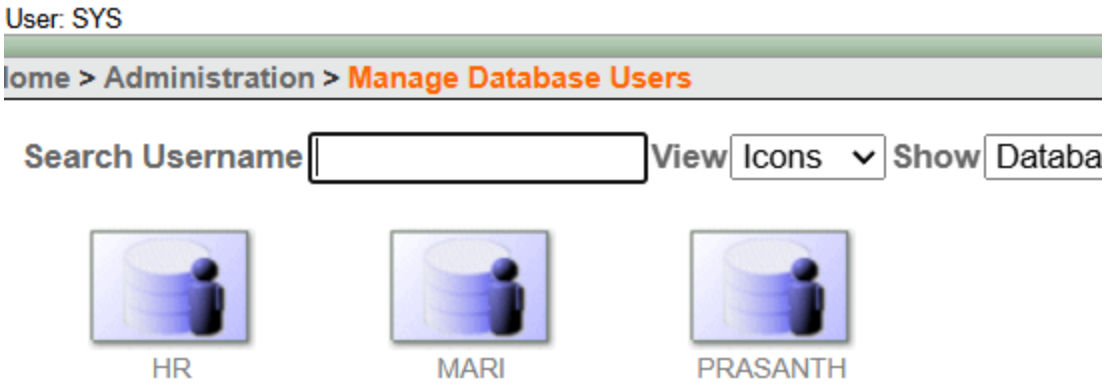
EMP_ID	NAME	HIRE_DATE	CUST_ID
101	prasanth	23-MAY-23	1
102	priya	24-MAY-23	2
104	thanga	26-JUN-23	4

3 rows returned in 0.00 seconds

So,successfully accessed hr table using user1(mari)

2.how to drop or delete user in database?

Here 3 users I have :

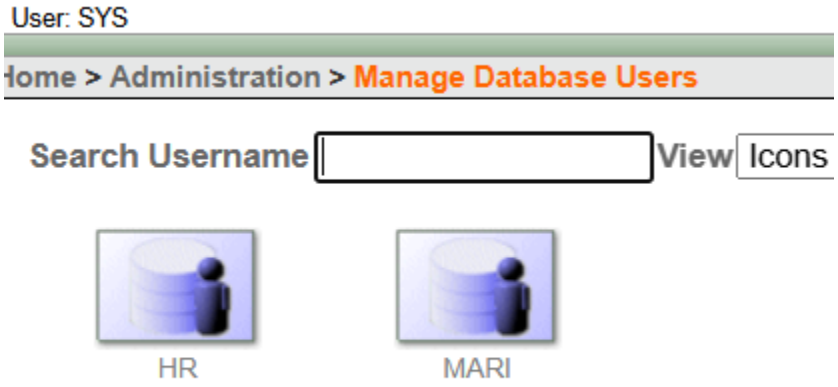


After disconnecting user(prasanth):

```
SQL> drop user prasanth cascade;

User dropped.
```

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:

```
EMPLOYEE_ID,FIRST_NAME,LAST_NAME,EMAIL,JOB_ID,SALARY,DEPARTMENT_ID
100,Steven,King,SKING,AD_PRES,24000,90
101,Neena,Kochhar,NKOCHHAR,AD_VP,17000,90
102,Lex,De Haan,LDEHAAN,AD_VP,17000,90
```

CSV FILE

```
C:\Users\Prashant>sqlldr hr/admin

control = D:\emp.ctl

SQL*Loader: Release 10.2.0.1.0 - Production on Wed May 28 23:49:34 2025

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Commit point reached - logical record count 3
Commit point reached - logical record count 4

C:\Users\Prashant>
```

```
CREATE TABLE emp (
  EMPLOYEE_ID      NUMBER(6),
  FIRST_NAME       VARCHAR2(30),
  LAST_NAME        VARCHAR2(20),
  EMAIL            VARCHAR2(100),
  JOB_ID           VARCHAR2(10),
  SALARY           NUMBER(8),
  DEPARTMENT_ID    NUMBER(4)
);

select * from emp;
```

Results | Explain | Describe | Saved SQL | History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	JOB_ID	SALARY	DEPARTMENT_ID
100	Steven	King	SKING	AD_PRES	24000	90
101	Neena	Kochhar	NKOCHHAR	AD_VP	17000	90
102	Lex	De Haan	LDEHAAN	AD_VP	17000	90

3 rows returned in 0.00 seconds CSV Export

Again adding using append command:

```
C:\Users\Prashant>sqlldr hr/admin

control = D:\emp.ctl

SQL*Loader: Release 10.2.0.1.0 - Production on Wed May 28 23:49:34 2025

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Commit point reached - logical record count 3
Commit point reached - logical record count 4

C:\Users\Prashant>sqlldr hr/admin

control = D:\emp.ctl

SQL*Loader: Release 10.2.0.1.0 - Production on Wed May 28 23:58:04 2025

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Commit point reached - logical record count 3
Commit point reached - logical record count 4
```

```
LOAD DATA
INFILE 'D:\emp1.csv'
INTO TABLE emp
APPEND
FIELDS TERMINATED BY ','
(EMPLOYEE_ID,
  FIRST_NAME,
  LAST_NAME,
  EMAIL,
  JOB_ID,
  SALARY,
  DEPARTMENT_ID)
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	JOB_ID	SALARY	DEPARTMENT_ID
103	Alexander	Hunold	AHUNOLD	IT_PROG	9000	60
104	Bruce	Ernst	BERNST	IT_PROG	6000	60
105	David	Austin	DAUSTIN	IT_PROG	4800	60
100	Steven	King	SKING	AD_PRES	24000	90
101	Neena	Kochhar	NKOCHHAR	AD_VP	17000	90
102	Lex	De Haan	LDEHAAN	AD_VP	17000	90

6 rows returned in 0.00 seconds CSV Export

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:

```
select * from user_tables

select * from employees;

select * from job_history


select * from departments;

SELECT e.employee_id,
       e.email,
       d.department_name,
       j.job_id
FROM employees e,
     departments d,
     job_history j
WHERE e.department_id = d.department_id
  AND d.department_id = j.department_id;
```

OUTPUT:

EMPLOYEE_ID	EMAIL	DEPARTMENT_NAME	JOB_ID
100	SKING	Executive	AD_ASST
100	SKING	Executive	AC_ACCOUNT
101	NKOCHHAR	Executive	AD_ASST
101	NKOCHHAR	Executive	AC_ACCOUNT
102	LDEHAAN	Executive	AD_ASST
102	LDEHAAN	Executive	AC_ACCOUNT
103	AHUNOLD	IT	IT_PROG
105	DAUSTIN	IT	IT_PROG
106	VPATABAL	IT	IT_PROG
107	DLORENTZ	IT	IT_PROG
120	MWEISS	Shipping	ST_CLERK
120	MWEISS	Shipping	ST_CLERK
121	AFRIPP	Shipping	ST_CLERK
121	AFRIPP	Shipping	ST_CLERK
122	PKAUFLIN	Shipping	ST_CLERK
122	PKAUFLIN	Shipping	ST_CLERK
123	SVOLLMAN	Shipping	ST_CLERK
123	SVOLLMAN	Shipping	ST_CLERK
124	KMOURGOS	Shipping	ST_CLERK
124	KMOURGOS	Shipping	ST_CLERK
More than 20 rows available. Increase rows selector to view more rows.			

20 rows returned in 0.00 seconds CSV Export

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;

```
SELECT * FROM ALL_SEQUENCES;

SELECT * FROM USER_SEQUENCES;

SELECT * FROM DBA_SEQUENCES;

SELECT sequence_name, min_value, max_value, increment_by, last_number
FROM user_sequences;
```

Results Explain Describe Saved SQL History

SEQUENCE_OWNER	SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG	ORDER_FLAG	CACHE_SIZE	LAST_NUMBER
SYS	UGROUP_SEQUENCE	0	999999999999999999999999999	1	N	Y	10	1
SYS	OBJECT_GRANT	1	999999999999999999999999999	1	N	Y	20	5441
SYS	SYSTEM_GRANT	1	999999999999999999999999999	1	N	Y	20	730
SYS	PROFNUM$	0	999999999999999999999999999	1	N	N	0	1
SYS	AUDSES$	1	2000000000	1	Y	N	20	122
SYS	SNAPSHOT_ID$	1	2147483647	1	N	N	20	1
SYS	SNAPSITE_ID$	1	4294967295	1	N	N	20	1
SYS	JOBSEQ	1	999999999	1	Y	N	20	41
SYS	RGROUPSEQ	1	999999999	1	Y	N	20	1
SYS	ORA_TQ_BASE$	1	4294967	1	Y	N	0	7
SYS	PARTITION_NAME$	1	999999999999999999999999999	1	N	N	20	21
SYS	CDC_SUBSCRIBE_SEQ$	1	999999999999999999999999999	1	N	N	20	1
SYS	CDC_RSID_SEQ$	1	999999999999999999999999999	1	N	Y	10000	1
SYS	LOG$SEQUENCE	0	999999999999999999999999999	1	N	Y	10	1
SYS	PSINDEX_SEQ$	1	18446744073709551615	1	N	N	1000	100
SYS	AWSEQ$	1	4294967295	1	N	N	0	1000
SYS	STREAMS$_CAPTURE_INST	1	4294967295	1	Y	N	0	1
SYS	APPLY$_SOURCE_OBJ_ID	1	999999999999999999999999999	1	N	N	0	1
SYS	APPLY$_DEST_OBJ_ID	1	999999999999999999999999999	1	N	N	0	1
SYS	APPLY$_ERROR_HANDLER_SEQUENCE	1	999999999999999999999999999	1	N	N	20	1

More than 20 rows available. Increase rows selector to view more rows.

20 rows returned in 0.00 seconds CSV Export

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;

```
SELECT * FROM USER_SEQUENCES;

SELECT * FROM DBA_SEQUENCES;

SELECT sequence_name, min_value, max_value, increment_by, last_number
FROM user_sequences;
```

Results Explain Describe Saved SQL History

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG	ORDER_FLAG	CACHE_SIZE	LAST_NUMBER
EMP_SEQ2	1	20	1	N	N	0	21
LOCATIONS_SEQ	1	9900	100	N	N	0	3300
DEPARTMENTS_SEQ	1	9990	10	N	N	0	280
EMPLOYEES_SEQ	1	999999999999999999999999999	1	N	N	0	207

4 rows returned in 0.00 seconds CSV Export

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;

```
SELECT * FROM DBA_SEQUENCES;

SELECT sequence_name, min_value, max_value, increment_by, last_number
FROM user_sequences;
```

Results | Explain | Describe | Saved SQL | History

SEQUENCE_OWNER	SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG	ORDER_FLAG	CACHE_SIZE	LAST_NUMBER
SYS	UGROUP_SEQUENCE	0	999999999999999999999999999	1	N	Y	10	1
SYS	OBJECT_GRANT	1	999999999999999999999999999	1	N	Y	20	5441
SYS	SYSTEM_GRANT	1	999999999999999999999999999	1	N	Y	20	730
SYS	PROFNUM$	0	999999999999999999999999999	1	N	N	0	1
SYS	AUDSES$	1	2000000000	1	Y	N	20	122
SYS	SNAPSHOT_ID$	1	2147483647	1	N	N	20	1
SYS	SNAPSITE_ID$	1	4294967295	1	N	N	20	1
SYS	JOBSEQ	1	999999999	1	Y	N	20	41
SYS	RGROUPSEQ	1	999999999	1	Y	N	20	1
SYS	ORA_TQ_BASE$	1	4294967	1	Y	N	0	7

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds CSV Export

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.**

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

```
CREATE SEQUENCE emp_seq2
  START WITH 1
  INCREMENT BY 1
  maxvalue 10
  NOCACHE
  NOCYCLE;

alter sequence emp_seq2
 maxvalue 20;
```

```
CREATE TABLE employees2 (
  emp_id NUMBER PRIMARY KEY,
  emp_name VARCHAR2(100)
);

INSERT INTO employees2 (emp_id, emp_name)
VALUES (emp_seq2.NEXTVAL, 'Alice');

INSERT INTO employees2 (emp_id, emp_name)
VALUES (emp_seq2.NEXTVAL, 'Bob');


SELECT emp_seq2.currval FROM employees2;

SELECT emp_seq2.nextval FROM employees2;
select * from employees2
```

Results Explain Describe Saved SQL History

EMP_ID	EMP_NAME
1	Alice
2	Bob
3	Bob
4	Bob
5	Bob
6	Bob
7	Bob
8	Bob
9	Bob
10	Bob
11	Alice
12	Alice
13	Alice
14	Alice
15	Alice
16	Alice
17	Alice
18	Alice
19	Alice
20	Alice

20 rows returned in 0.00 seconds CSV Export

Assignment-8

GROUP BY

1. HOW MANY EMPLOYEE WORKING DEPARTMENT WISE?

```
select department_id, count(*) as emp_count
from employees
group by department_id;
```

Results | Explain | Describe | Saved SQL | History

DEPARTMENT_ID	EMP_COUNT
100	6
30	6
-	1
90	3
20	2
70	1
110	2
50	45
80	34
40	1
60	4
10	1

12 rows returned in 0.02 seconds CSV Export

based on country_id in location table:

```
select * from locations

select country_id, count(*) as location_id
from locations
group by country_id;
```

Results | Explain | Describe | Saved SQL | History

COUNTRY_ID	LOCATION_ID
US	4
SG	1
CA	2
CH	2
IT	2
MX	1
CN	1
DE	1
JP	2
IN	1
AU	1
UK	3
BR	1
NL	1

14 rows returned in 0.00 seconds CSV Export

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.

```
select * from user_views
```

Results | Explain | Describe | Saved SQL | History

VIEW_NAME	TEXT_LENGTH	TEXT
EMP_DETAILS_VIEW	538	SELECT e.employee_id, e.job_id, e.manager_id, e.department_id, d.location_id, l.country_id, e.first_name, e.last_name, e.salary, e.commission_pct, d.department_name, j.job_title, l.city, l.state_province, c.country_name, r.region_name FROM employees e, departments d, jobs j, locations l, countries c, regions r WHERE e.department_id = d.department_id AND d.location_id = l.location_id AND l.country_id = c.country_id AND c.region_id = r.region_id AND j.job_id = e.job_id WITH READ ONLY
EMPV1	196	select "EMPLOYEE_ID","FIRST_NAME","LAST_NAME","EMAIL","PHONE_NUMBER","HIRE_DATE","JOB_ID","SALARY","COMMISSION_PCT","MANAGER_ID","DEPARTMENT_ID" from employees where salary between 10000 and 20000
DEPT10_EMPLOYEES	81	SELECT employee_id, salary,department_id FROM employees WHERE department_id = 100

3 rows returned in 0.01 seconds CSV Export

2. ALL_VIEWS:

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

```
select * from all_views
```

Results Explain Describe Saved SQL History

OWNER	VIEW_NAME	TEXT_LENGTH	
SYS	V_$MAP_LIBRARY	160	select "LIB_IDX","LIB_NAME","VENDOR_NAME","PROTOCOL_NUM","VERSION_NUM","PATH_NAME","MAP_FILE","FILE_CFGID","MAP_ELEM","ELEM_CFGID","MAP_SYNC" from v$map_library
SYS	V_$MAP_FILE	140	select "FILE_MAP_IDX","FILE_CFGID","FILE_STATUS","FILE_NAME","FILE_TYPE","FILE_STRUCTURE","FILE_SIZE","FILE_NEXTS","LIB_IDX" from v$map_file
SYS	V_$MAP_FILE_EXTENT	117	select "FILE_MAP_IDX","EXT_NUM","EXT_ELEM_OFF","EXT_SIZE","EXT_FILE_OFF","EXT_TYPE","ELEM_IDX" from v$map_file_extent
SYS	V_$MAP_ELEMENT	138	select "ELEM_NAME","ELEM_IDX","ELEM_CFGID","ELEM_TYPE","ELEM_SIZE","ELEM_NSUBELEM","ELEM_DESCR","STRIPE_SIZE","LIB_IDX" from v$map_element
SYS	V_$MAP_EXT_ELEMENT	187	select "ELEM_IDX","NUM_ATTRB","ATTRB1_NAME","ATTRB1_VAL","ATTRB2_NAME","ATTRB2_VAL","ATTRB3_NAME","ATTRB3_VAL","ATTRB4_NAME","ATTRB4_VAL","ATTRB5_NAME","ATTRB5_VAL" from v$map_ext_element
SYS	V_$MAP_COMP_LIST	174	select "ELEM_IDX","NUM_COMP","COMP1_NAME","COMP1_VAL","COMP2_NAME","COMP2_VAL","COMP3_NAME","COMP3_VAL","COMP4_NAME","COMP4_VAL","COMP5_NAME","COMP5_VAL" from v$map_comp_list
SYS	V_$MAP_SUBELEMENT	100	select "CHILD_IDX","PARENT_IDX","SUB_NUM","SUB_SIZE","ELEM_OFFSET","SUB_FLAGS" from v$map_subelement
SYS	V_$MAP_FILE_IO_STACK	179	select "FILE_MAP_IDX","DEPTH","ELEM_IDX","CU_SIZE","STRIDE","NUM_CU","ELEM_OFFSET","FILE_OFFSET","DATA_TYPE","PARITY_POS","PARITY_PERIOD","ID","PARENT_ID" from v$map_file_io_stack
SYS	V_$SQL_REDIRECTION	211	select "ADDRESS","PARENT_HANDLE","HASH_VALUE","SQL_ID","CHILD_NUMBER","PARSING_USER_ID","PARSING_SCHEMA_ID","COMMAND_TYPE","REASON","ERROR_CODE","POSITION","SQL_TEXT_PIECE","ERROR_M
SYS	V_$SQL_PLAN	528	select "ADDRESS","HASH_VALUE","SQL_ID","PLAN_HASH_VALUE","CHILD_ADDRESS","CHILD_NUMBER","TIMESTAMP","OPERATION","OPTIONS","OBJECT_NODE","OBJECT#","OBJECT_OWNER","OBJECT_NAME","OBJECT_ALIA from v$sql_plan

More than 10 rows available. Increase rows selector to view more rows.

3. DBA_VIEWS:

Lists all views in the entire database, across all schemas. *(requires DBA privileges)*

```
select * from dba_views;
```

Results Explain Describe Saved SQL History

OWNER	VIEW_NAME	TEXT_LENGTH	
SYS	V_$MAP_LIBRARY	160	select "LIB_IDX","LIB_NAME","VENDOR_NAME","PROTOCOL_NUM","VERSION_NUM","PATH_NAME","MAP_FILE","FILE_CFGID","MAP_ELEM","ELEM_CFGID","MAP_SYNC" from v$map_library
SYS	V_$MAP_FILE	140	select "FILE_MAP_IDX","FILE_CFGID","FILE_STATUS","FILE_NAME","FILE_TYPE","FILE_STRUCTURE","FILE_SIZE","FILE_NEXTS","LIB_IDX" from v$map_file
SYS	V_$MAP_FILE_EXTENT	117	select "FILE_MAP_IDX","EXT_NUM","EXT_ELEM_OFF","EXT_SIZE","EXT_FILE_OFF","EXT_TYPE","ELEM_IDX" from v$map_file_extent
SYS	V_$MAP_ELEMENT	138	select "ELEM_NAME","ELEM_IDX","ELEM_CFGID","ELEM_TYPE","ELEM_SIZE","ELEM_NSUBELEM","ELEM_DESCR","STRIPE_SIZE","LIB_IDX" from v$map_element
SYS	V_$MAP_EXT_ELEMENT	187	select "ELEM_IDX","NUM_ATTRB","ATTRB1_NAME","ATTRB1_VAL","ATTRB2_NAME","ATTRB2_VAL","ATTRB3_NAME","ATTRB3_VAL","ATTRB4_NAME","ATTRB4_VAL","ATTRB5_NAME","ATTRB5_VAL" from v$map_ext_element
SYS	V_$MAP_COMP_LIST	174	select "ELEM_IDX","NUM_COMP","COMP1_NAME","COMP1_VAL","COMP2_NAME","COMP2_VAL","COMP3_NAME","COMP3_VAL","COMP4_NAME","COMP4_VAL","COMP5_NAME","COMP5_VAL" from v$map_comp_list
SYS	V_$MAP_SUBELEMENT	100	select "CHILD_IDX","PARENT_IDX","SUB_NUM","SUB_SIZE","ELEM_OFFSET","SUB_FLAGS" from v$map_subelement
SYS	V_$MAP_FILE_IO_STACK	179	select "FILE_MAP_IDX","DEPTH","ELEM_IDX","CU_SIZE","STRIDE","NUM_CU","ELEM_OFFSET","FILE_OFFSET","DATA_TYPE","PARITY_POS","PARITY_PERIOD","ID","PARENT_ID" from v$map_file_io_stack
SYS	V_$SQL_REDIRECTION	211	select "ADDRESS","PARENT_HANDLE","HASH_VALUE","SQL_ID","CHILD_NUMBER","PARSING_USER_ID","PARSING_SCHEMA_ID","COMMAND_TYPE","REASON","ERROR_CODE","POSITION","SQL_TEXT_PIECE","ERROR_
SYS	V_$SQL_PLAN	528	select "ADDRESS","HASH_VALUE","SQL_ID","PLAN_HASH_VALUE","CHILD_ADDRESS","CHILD_NUMBER","TIMESTAMP","OPERATION","OPTIONS","OBJECT_NODE","OBJECT#","OBJECT_OWNER","OBJECT_NAME","OBJECT_ALI from v$sql_plan

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds CSV Export

Column Name	Description
VIEW_NAME	Name of the view
TEXT	SQL query text that defines the view

Column Name	Description
OWNER	Owner of the view (only in ALL_ / DBA_)

Example: Show All Views and Their SQL Definition:

```
SELECT view_name, text
FROM user_views;
```

Results Explain Describe Saved SQL History

VIEW_NAME	TEXT
EMP_DETAILS_VIEW	SELECT e.employee_id, e.job_id, e.manager_id, e.department_id, d.location_id, l.country_id, e.first_name, e.last_name, e.salary, e.commission_pct, d.department_name, j.job_title, l.city, l.state_province, c.country_name, r.region_name FROM employees e, departments d, jobs j, locations l, countries c, regions r WHERE e.department_id = d.department_id AND d.location_id = l.location_id AND l.country_id = c.country_id AND c.region_id = r.region_id AND j.job_id = e.job_id WITH READ ONLY
EMPV1	select "EMPLOYEE_ID","FIRST_NAME","LAST_NAME","EMAIL","PHONE_NUMBER","HIRE_DATE","JOB_ID","SALARY","COMMISSION_PCT","MANAGER_ID","DEPARTMENT_ID" from employees where salary between 10000 and 20000
DEPT10_EMPLOYEES	SELECT employee_id, salary,department_id FROM employees WHERE department_id = 100

3 rows returned in 0.00 seconds CSV Export

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

```
SELECT dbms_metadata.get_ddl('VIEW', 'DEPT10_EMPLOYEES') FROM dual;
```

Results Explain Describe Saved SQL History

DBMS_METADATA.GET_DDL('VIEW','DEPT10_EMPLOYEES')
CREATE OR REPLACE FORCE VIEW "HR"."DEPT10_EMPLOYEES" ("EMPLOYEE_ID", "SALARY", "DEPARTMENT_ID") AS SELECT employee_id, salary,department_id FROM employees WHERE department_id = 100

1 rows returned in 0.09 seconds CSV Export

4. Get the Columns of a View

Use the **USER_TAB_COLUMNS**, **ALL_TAB_COLUMNS**, or **DBA_TAB_COLUMNS** metadata views:

```
select column_name, data_type, data_length, table_name
from user_tab_columns
where table_name = 'EMPV1';
```

Results Explain Describe Saved SQL History

COLUMN_NAME	DATA_TYPE	DATA_LENGTH	TABLE_NAME
EMPLOYEE_ID	NUMBER	22	EMPV1
FIRST_NAME	VARCHAR2	20	EMPV1
LAST_NAME	VARCHAR2	25	EMPV1
EMAIL	VARCHAR2	25	EMPV1
PHONE_NUMBER	VARCHAR2	20	EMPV1
HIRE_DATE	DATE	7	EMPV1
JOB_ID	VARCHAR2	10	EMPV1
SALARY	NUMBER	22	EMPV1
COMMISSION_PCT	NUMBER	22	EMPV1
MANAGER_ID	NUMBER	22	EMPV1
More than 10 rows available. Increase rows selector to view more rows.			

10 rows returned in 0.01 seconds CSV Export

```
select owner, column_name, data_type
from all_tab_columns
where table_name = 'DEPT10_EMPLOYEES';
```

Results | Explain | Describe | Saved SQL | History

OWNER	COLUMN_NAME	DATA_TYPE
HR	EMPLOYEE_ID	NUMBER
HR	SALARY	NUMBER
HR	DEPARTMENT_ID	NUMBER

3 rows returned in 0.00 seconds CSV Export

```
select owner, column_name, data_type
from dba_tab_columns
where table_name = 'DEPT10_EMPLOYEES';
```

Results | Explain | Describe | Saved SQL | History

OWNER	COLUMN_NAME	DATA_TYPE
HR	EMPLOYEE_ID	NUMBER
HR	SALARY	NUMBER
HR	DEPARTMENT_ID	NUMBER

3 rows returned in 0.01 seconds CSV Export

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:

```
select view_name,
       DBMS_METADATA.GET_DDL('VIEW', view_name) AS view_ddl
from user_views
where DBMS_METADATA.GET_DDL('VIEW', view_name) LIKE '%EMAIL%';
```

Results | Explain | Describe | Saved SQL | History

VIEW_NAME	VIEW_DDL
EMPV1	CREATE OR REPLACE FORCE VIEW "HR"."EMPV1" ("EMPLOYEE_ID", "FIRST_NAME", "LAST_NAME", "EMAIL", "PHONE_NUMBER", "HIRE_DATE", "JOB_ID", "SALARY", "COMMISSION_PCT", "MANAGER_ID", "DEPARTMENT_ID") AS select "EMPLOYEE_ID","FIRST_NAME","LAST_NAME","EMAIL","PHONE_NUMBER","HIRE_DATE","JOB_ID","SALARY","COMMISSION_PCT","MANAGER_ID","DEPARTMENT_ID" from employees where salary between 10000 and 20000

1 rows returned in 0.19 seconds CSV Export

```
select view_name,
       DBMS_METADATA.GET_DDL('VIEW', view_name) AS view_ddl
from user_views
where DBMS_METADATA.GET_DDL('VIEW', view_name) LIKE '%EMPLOYEE_ID%';
```

Results | Explain | Describe | Saved SQL | History

VIEW_NAME	VIEW_DDL
EMP_DETAILS_VIEW	CREATE OR REPLACE FORCE VIEW "HR"."EMP_DETAILS_VIEW" ("EMPLOYEE_ID", "JOB_ID", "MANAGER_ID", "DEPARTMENT_ID", "LOCATION_ID", "COUNTRY_ID", "FIRST_NAME", "LAST_NAME", "SALARY", "COMMISSION_PCT", "DEPARTMENT_NAME", "JOB_TITLE", "CITY", "STATE_PROVINCE", "COUNTRY_NAME", "REGION_NAME") AS SELECT e.employee_id, e.job_id, e.manager_id, e.department_id, d.location_id, l.country_id, e.first_name, e.last_name, e.salary, e.commission_pct, d.department_name, j.job_title, l.city, l.state_province, c.country_name, r.region_name FROM employees e, departments d, jobs j, locations l, countries c, regions r WHERE e.department_id = d.department_id AND d.location_id = l.location_id AND l.country_id = c.country_id AND c.region_id = r.region_id AND j.job_id = e.job_id WITH READ ONLY
EMPV1	CREATE OR REPLACE FORCE VIEW "HR"."EMPV1" ("EMPLOYEE_ID", "FIRST_NAME", "LAST_NAME", "EMAIL", "PHONE_NUMBER", "HIRE_DATE", "JOB_ID", "SALARY", "COMMISSION_PCT", "MANAGER_ID", "DEPARTMENT_ID") AS select "EMPLOYEE_ID","FIRST_NAME","LAST_NAME","EMAIL","PHONE_NUMBER","HIRE_DATE","JOB_ID","SALARY","COMMISSION_PCT","MANAGER_ID","DEPARTMENT_ID" from employees where salary between 10000 and 20000
DEPT10_EMPLOYEES	CREATE OR REPLACE FORCE VIEW "HR"."DEPT10_EMPLOYEES" ("EMPLOYEE_ID", "SALARY", "DEPARTMENT_ID") AS SELECT employee_id, salary,department_id FROM employees WHERE department_id = 100

3 rows returned in 0.28 seconds CSV Export

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

Purpose	Metadata View
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:

```
)
organization external (
    type oracle_loader
    default directory emp_dtails
    access parameters (
        records delimited by newline
        fields terminated by ','
        missing field values are null
        (
            employee_id, first_name, last_name, email, phone_number,
            hire_date DATE 'YYYY-MM-DD',
            job_id, salary, commission_pct, manager_id, department_id,
            address, city, state, postal_code, country,
            birth_date DATE 'YYYY-MM-DD',
            gender, marital_status CHAR(25)
        )
    )
    location (
        'employees_details.csv',
        'employees_details_next.txt'
    )
)
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	ADDRESS	CITY	STATE	POSTAL_CODE	COUNTRY	BIRTH_DATE	GENDER	MARITAL_STATUS
1001	John	Doe	john.doe@example.com	555-1234	15-JAN-20	DEV001	60000	.05	2001	10	123 Main St	New York	NY	10001	USA	10-JUN-85	M	Single
1002	Rachel	Green	rachel.green@example.com	555-6661	11-FEB-21	HR011	55000	.03	2011	110	123 Fashion Ave	New York	NY	10018	USA	05-MAY-90	F	Single

2 rows returned in 0.02 seconds CSV Export

Selecting all directories:

```
select * from all_directories
```

Results | Explain | Describe | Saved SQL | History

OWNER	DIRECTORY_NAME	DIRECTORY_PATH
SYS	DATA_PUMP_DIR	C:\oraclexe\app\oracle\admin\XE\dpdump\
SYS	ORACLECLRDIR	C:\oraclexe\app\oracle\product\10.2.0\server\bin\clr
SYS	EMP_DTAILS	V:\

3 rows returned in 0.00 seconds CSV Export

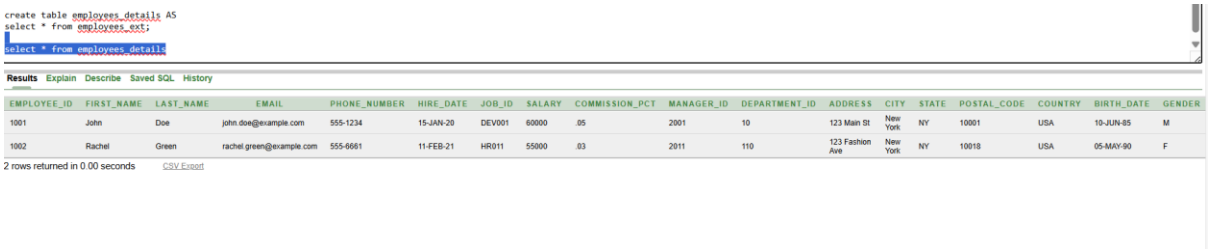
When working on external tables the most commonly used format is CSV

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:

```
SELECT * FROM USER_SYS_PRIVS WHERE PRIVILEGE LIKE '%DIRECTORY%'
```

Results Explain Describe Saved SQL History

USERNAME	PRIVILEGE	ADMIN_OPTION
HR	CREATE ANY DIRECTORY	NO

1 rows returned in 0.00 seconds CSV Export

Assignment-11

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

```
select * from employees where salary = (
select max(salary)
from employees
where salary < (
    select max(salary) from employees
)
);
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	-	100	90

2 rows returned in 0.00 seconds CSV Export

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

```
select *
from employees
where (department_id, salary) in (
    select department_id, max(salary)
    from employees
    group by department_id
);
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-94	FI_MGR	12000	-	101	100
114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-94	PU_MAN	11000	-	100	30
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	-	-	90
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	13000	-	100	20
204	Hermann	Baer	HBAER	515.123.8888	07-JUN-94	PR_REP	10000	-	101	70
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	12000	-	101	110
121	Adam	Fripp	AFRIPP	650.123.2234	10-APR-97	ST_MAN	8200	-	100	50
145	John	Russell	JRUSSEL	011.44.1344.429268	01-OCT-96	SA_MAN	14000	.4	100	80
203	Susan	Mavris	SMAVRIS	515.123.7777	07-JUN-94	HR_REP	6500	-	101	40
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	-	102	60
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	-	101	10

11 rows returned in 0.00 seconds CSV Export

3. current salary of employees, max_salary and min_salary of employees and display first_name,salary,max_salary,min_salary?

```
select first_name, salary as current_salary,
    (select max(salary) from employees) as max_salary,
    (select min(salary) from employees) as min_salary
from employees;
```

Results Explain Describe Saved SQL History

FIRST_NAME	CURRENT_SALARY	MAX_SALARY	MIN_SALARY
Steven	24000	24000	2100
Neena	17000	24000	2100
Lex	17000	24000	2100
Alexander	9000	24000	2100
David	4800	24000	2100
Valli	4800	24000	2100
Diana	4200	24000	2100
Nancy	12000	24000	2100
Daniel	9000	24000	2100
John	8200	24000	2100
Ismael	7700	24000	2100
Jose Manuel	7800	24000	2100
Luis	6900	24000	2100
Den	11000	24000	2100
Alexander	3100	24000	2100
More than 15 rows available. Increase rows selector to view more rows.			

15 rows returned in 0.00 seconds CSV Export

Assignment-12

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

```
select lpad(rpad(first_name,length(first_name)+3,'*'),
   length(first_name)+6,'*')
   as lr_pad
from employees;
```

Results | Explain | Describe | Saved SQL | History

LR_PAD
Ellen
Sundar
Mozhe
David
Hermann
Shelli
Amit
Elizabeth
Sarah
David
More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds CSV Export

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)?

```
create table emp1 (
    name varchar2(20)
);

insert into emp1 (name) values ('tamara');
insert into emp1 (name) values ('arabella');
insert into emp1 (name) values ('kamala');
insert into emp1 (name) values ('amaranta');
insert into emp1 (name) values ('anastasia');
insert into emp1 (name) values ('amala');

select name,substr(name,
          instr(name, 'a', 1, 2) + 1,
          instr(name, 'a', 1, 3) - instr(name, 'a', 1, 2) - 1
       ) as btw_2nd_3rd
from emp1;
```

Results Explain Describe Saved SQL History

NAME	BTW_2ND_3RD
tamara	r
kamala	l
arabella	bell
amaranta	r
anastasia	st
amala	l

6 rows returned in 0.00 seconds CSV Export

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:**

```
select department_id,
case
  when department_id = 90 then 'AUDIT'
  when department_id = 60 then 'HR'
  else 'others'
  end as c_w
from employees;
```

Results | Explain | Describe | Saved SQL | History

DEPARTMENT_ID	C_W
90	AUDIT
90	AUDIT
90	AUDIT
60	HR
60	HR
60	HR
60	HR
100	others
100	others
100	others
100	others
100	others
100	others
30	others
30	others
30	others

2. print the first_date of the month

```
select to_char(last_day(sysdate - 31) + 1, 'day dd-mon-yy') as first_date from dual;
```

Results Explain Describe Saved SQL History

FIRST_DATE
sunday 01-jun-25

1 rows returned in 0.00 seconds CSV Export