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
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
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