

# Creating Other Schema Objects

# Objectives

- After completing this lesson, you should be able to do the following:
  - Create simple and complex views
  - Retrieve data from views

# Database Objects

Object	Description
Table	Basic unit of storage; composed of rows
View	Logically represents subsets of data from one or more tables
Sequence	Generates numeric values
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

# What Is a View?

EMPLOYEES table

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_FRES	2400
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	1700
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	1700
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	900
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	600
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-98	IT_PROG	420
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	580
141	Trenna	Rae	TRAJIS	650.121.8009	17-OCT-95	ST_CLERK	350
142	Curtis	Denes	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	310
143	Randall	Mateo	RMATEO	650.121.2074	10-MAR-98	ST_CLERK	260
149	Zlotkey				24-MAY-96	ST_CLERK	250
174	Abel				24-JAN-00	SA_MAN	1050
176	Taylor				24-MAY-96	SA_REP	1100
177	Rudney				24-MAR-98	SA_REP	860
178	Kimberly	Grant	KGRANT	611.44.1044.429203	24-MAY-99	SA_REP	700
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	440
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	1300
202	Pat	Fay	PFAY	603.123.6666	17-AUG-97	MK_REP	600
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	1200
206	William	Gietz	WGIEZT	515.123.8181	07-JUN-94	AC_ACCOUNT	830

20 rows selected.

# What is a view?

You can present logical subsets or combinations of data by creating views of tables.

A **view** is a **logical table** based on a table or another view.

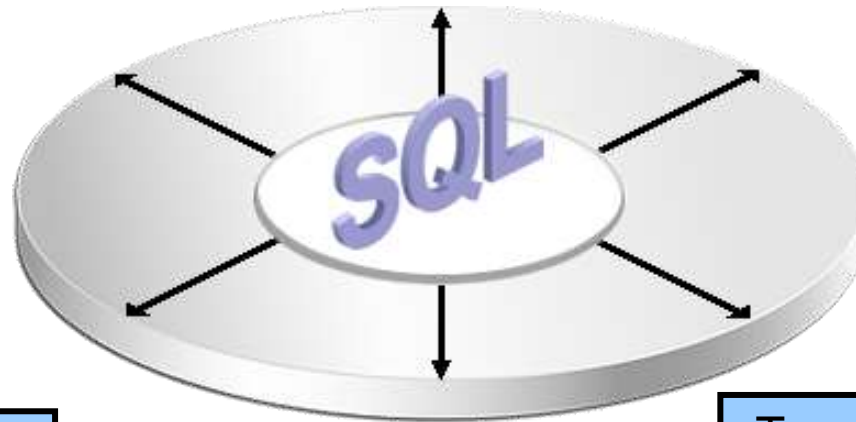
A view contains no data of its own but is **like a window** through which data from tables **can be viewed or changed**.

The tables on which a view is based are called **base tables**. The view is stored as a SELECT statement in the data dictionary.

# Advantages of Views

To restrict  
data access

To make complex  
queries easy



To provide data  
independence

To present different  
views of the same  
data

# Simple Views and Complex Views

Feature	Simple Views	Complex Views
Number of tables	One	One or more
Contain functions	No	Yes
Contain groups of data	No	Yes
DML operations through a view	Yes	Not always

# Creating a View

- You embed a subquery in the CREATE VIEW statement:

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view  
  [(alias[, alias]...)]  
  AS subquery  
[WITH CHECK OPTION [CONSTRAINT constraint]]  
[WITH READ ONLY [CONSTRAINT constraint]];
```

- The subquery can contain complex SELECT syntax.



# Creating a View

- Create the EMPVU80 view, which contains details of employees in department 80:

```
CREATE VIEW    empvu80
  AS SELECT    employee_id, last_name, salary
    FROM      employees
    WHERE      department_id = 80;
```

**View created.**

- Describe the structure of the view by using the SQL\*Plus DESCRIBE command:

```
DESCRIBE empvu80
```

# Creating a View

- Create a view by using column aliases in the subquery:

```
CREATE VIEW    salvu50
  AS SELECT    employee_id ID_NUMBER, last_name NAME,
              salary*12 ANN_SALARY
  FROM        employees
  WHERE       department_id = 50;
View created.
```

## Or in the CREATE statement:

```
CREATE OR REPLACE VIEW salvu50(ID_NUMBER, NAME, ANN_SALARY)
  AS SELECT    employee_id, last_name, salary*12
  FROM        employees
  WHERE       department_id = 50;
```

# Retrieving Data from a View

```
SELECT *  
FROM salvu50;
```

ID_NUMBER	NAME	ANN_SALARY
124	Mourgos	69600
141	Rajs	42000
142	Davies	37200
143	Matos	31200
144	Vargas	30000

# Modifying a View

- Modify the EMPVU80 view by using a CREATE OR REPLACE VIEW clause. Add an alias for each column name:

```
CREATE OR REPLACE VIEW empvu80
  (id_number, name, sal, department_id)
AS SELECT  employee_id, first_name || ' '
           || last_name, salary, department_id
  FROM      employees
 WHERE     department_id = 80;
View created.
```

- Column aliases in the CREATE OR REPLACE VIEW clause are listed in the same order as the columns in the subquery.



# Creating a Complex View

- Create a complex view that contains **group functions** to display values from two tables:

```
CREATE OR REPLACE VIEW dept_sum_vu
  (name, minsal, maxsal, avgsal)
AS SELECT    d.department_name, MIN(e.salary),
             MAX(e.salary), AVG(e.salary)
  FROM      employees e JOIN departments d
  ON        (e.department_id = d.department_id)
  GROUP BY  d.department_name;
```

**View created.**

# Rules for Performing DML Operations on a View

- You can usually perform DML operations on simple views. 
- You **cannot remove** a row **if the view contains** the following:
  - Group functions
  - A `GROUP BY` clause
  - The `DISTINCT` keyword
  - The pseudocolumn `ROWNUM` keyword

# Rules for Performing DML Operations on a View

- You **cannot modify** data in a view **if it contains**:
  - Group functions
  - A `GROUP BY` clause
  - The `DISTINCT` keyword
  - The pseudocolumn `ROWNUM` keyword
  - Columns defined by **expressions**

# Rules for Performing DML Operations on a View

- You **cannot add** data through a view **if the view includes**:
  - Group functions
  - A `GROUP BY` clause
  - The `DISTINCT` keyword
  - The pseudocolumn `ROWNUM` keyword
  - Columns defined by expressions
  - **NOT NULL columns** in the base tables that are not selected by the view



# Using the WITH CHECK OPTION Clause

- You can **ensure** that DML operations performed on the view **stay in the domain of the view** by using the **WITH CHECK OPTION** clause:

```
CREATE OR REPLACE VIEW empvu20
AS SELECT      *
   FROM        employees
   WHERE       department_id = 20
   WITH CHECK OPTION CONSTRAINT empvu20_ck ;
```

**View created.**

- Any attempt to change the department number for any row in the view fails because it violates the **WITH CHECK OPTION** constraint.

# Denying DML Operations

- You can **ensure** that **no DML operations occur** by adding the **WITH READ ONLY** option to your view definition.
- Any attempt to perform a DML operation on any row in the view results in an Oracle server error.



# Denying DML Operations

```
CREATE OR REPLACE VIEW empvu10  
    (employee_number, employee_name, job_title)  
AS SELECT      employee_id, last_name, job_id  
    FROM        employees  
    WHERE       department_id = 10  
    WITH READ ONLY ;
```

**View created.**

# Removing a View

- You can remove a view without losing data because a view is based on underlying tables in the database.

```
DROP VIEW view;
```

```
DROP VIEW empvu80;  
View dropped.
```