

EXERCISE-11

CREATING VIEWS

After the completion of this exercise, students will be able to do the following:

- Describe a view
- Create, alter the definition of, and drop a view
- Retrieve data through a view
- Insert, update, and delete data through a view
- Create and use an inline view

View

A view is a logical table based on a table or another view. A view contains no data 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.

Advantages of Views

- To restrict data access
- To make complex queries easy
- To provide data independence
- To present different views of the same data

Classification of views

1. Simple view
2. Complex view

Feature	Simple	Complex
No. of tables	One	One or more
Contains functions	No	Yes
Contains groups of data	No	Yes
DML operations thr' view	Yes	Not always

Creating a view

Syntax

CREATE OR REPLACE FORCE/NOFORCE VIEW *view_name* AS Subquery WITH CHECK
OPTION CONSTRAINT *constraint* WITH READ ONLY CONSTRAINT *constraint*;

FORCE - Creates the view regardless of whether or not the base tables exist.

NOFORCE - Creates the view only if the base table exist.

WITH CHECK OPTION CONSTRAINT-specifies that only rows accessible to the view can be inserted or updated.

WITH READ ONLY CONSTRAINT-ensures that no DML operations can be performed on the view.

Example: 1 (Without using Column aliases)

Create a view EMPVU80 that contains details of employees in department 80.

Example 2:

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

Example:1 (Using column aliases)

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

Retrieving data from a view

Example:

```
SELECT * from salvu50;
```

Modifying a view

A view can be altered without dropping, re-creating.

Example: (Simple view)

Modify the EMPVU80 view by using CREATE OR REPLACE.

```
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;
```

Example: (complex view)

```
CREATE 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, department d  
WHERE e.department_id=d.department_id  
GROUP BY d.department_name;
```

Rules for performing DML operations on view

- Can perform operations on simple views
- Cannot remove a row if the view contains the following:
 - Group functions
 - Group By clause
 - Distinct keyword
- Cannot modify data in a view if it contains
 - Group functions
 - Group By clause
 - Distinct keyword
- Columns contain by expressions
- Cannot add data thr' a view if it contains
 - Group functions

- Group By clause
- Distinct keyword
- Columns contain by expressions
- NOT NULL columns in the base table that are not selected by the view

Example: (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;
```

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

Example – (Execute this and note the error)

```
UPDATE empvu20 SET department_id=10 WHERE employee_id=201;
```

Denying DML operations

Use of WITH READ ONLY option.

Any attempt to perform a DML on any row in the view results in an oracle server error.

Try this code:

```
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;
```

Find the Solution for the following:

1. Create a view called EMPLOYEE_VU based on the employee numbers, employee names and department numbers from the EMPLOYEES table. Change the heading for the employee name to EMPLOYEE.

Creating view employee_vu as select emp_id, f-name || " " || l-name as emp, dept_id from emp;

2. Display the contents of the EMPLOYEES_VU view.

*Select * from employee_vu;*

3. Select the view name and text from the USER_VIEWS data dictionary views.

Select view_name, text from user_views where view_name = 'EMPLOYEE_VU'
'employee_nu';

4. Using your EMPLOYEES_VU view, enter a query to display all employees names and department.

Select employee, dept_id from emp_vu;

5. Create a view named DEPT50 that contains the employee number, employee last names and department numbers for all employees in department 50. Label the view columns EMPNO, EMPLOYEE and DEPTNO. Do not allow an employee to be reassigned to another department through the view.

Create view dept50 as select employee_id as empno;

6. Display the structure and contents of the DEPT50 view.

{update dept50; set deptno = 80 where emp = 'matos';

Attempt to reassign Matos to department 80.

{close dept50; select * from dept50;

8. Create a view called SALARY_VU based on the employee last names, department names, salaries, and salary grades for all employees. Use the Employees, DEPARTMENTS and JOB_GRADE tables. Label the column Employee, Department, salary, and Grade respectively.

Create view Salary_VU as Select e.last-name as emp,
d.dept-name as dept, e.salary &, job-grade level &
grade from employees e join dept d using (dept_id)
join job-grade j on e.salary between j.lowest-sal
and j.highest-sal;

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	BPL

Practice Problems -I

Join Clauses

Use the Oracle database for problems 1-6.

1. Join the Oracle database locations and departments table using the location_id column. Limit the results to location 1400 only.

Select * From locations l Join departments d ON l.location_id = d.location_id
d.location_id where l.location_id = 1400;

2. Join DJs on Demand d_play_list_items, d_track_listings, and d_cds tables with the JOIN USING syntax. Include the song ID, CD number, title, and comments in the output.

Select t.song_id, c.cd_number, c.title, c.comments From
d_play_list_items i Join d-track_listings t Using (Song_id) JOIN d_cds c
using (cd-number);

3. Display the city, department name, location ID, and department ID for departments 10, 20, and 30 for the city of Seattle.

Select l.city, d.department_name, l.location_id, d.department_id
From locations l Join departments d ON l.location_id = d.location_id
Where d.department_id IN(10, 20, 30) AND l.city = 'Seattle';

4. Display country name, region ID, and region name for Americas.

Select c.country_name, r.region_id, r.region_name From countries c
JOIN regions r Using (region_id) where r.region_name =
'Americas';

5. Write a statement joining the employees and jobs tables. Display the first and last names, hire date, job id, job title, and maximum salary. Limit the query to those employees who are in jobs that can earn more than \$12,000.

Select e.first_name, e.last_name, e.hire_date, j.job_id, j.job_title
From employees e JOIN jobs j ON e.job_id = j.job_id
Where j.max_salary > 12000;

Inner versus Outer Joins

Use the Oracle database for problems 1-7.

1. Return the first name, last name, and department name for all employees including those employees not assigned to a department.

Select e.first-name, e.last-name, d.department-name FROM employees
e LEFT outer JOIN departments d ON e.department-id = d.department-id;

2. Return the first name, last name, and department name for all employees including those departments that do not have an employee assigned to them.

Select e.first-name, e.last-name, d.department-name FROM employees
e RIGHT outer JOIN departments d ON e.department-id
= d.department-id;

3. Return the first name, last name, and department name for all employees including those departments that do not have an employee assigned to them and those employees not assigned to a department.

Select e.first-name, e.last-name, d.department-name FROM employees
e FULL outer JOIN departments d ON e.department-id = d.department-id;

4. Create a query of the DJs on Demand database to return the first name, last name, event date, and description of the event the client held. Include all the clients even if they have not had an event scheduled.

Select c.first-name, c.last-name, e.event-date, e.description FROM
c-clients e LEFT outer JOIN d-events e ON c.client-number:
e.client-number;

5. Using the Global Fast Foods database, show the shift description and shift assignment date even if there is no date assigned for each shift description.

Select s.shift-description, a.assignment-date FROM f-shifts s
LEFT outer JOIN f-shift-assignment a ON s.shift-id = a.shift-id;

Sell Joins and Hierarchical Queries

For each problem, use the Oracle database.

1. Display the employee's last name and employee number along with the manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.

Select e.last-name AS "Employee", e.employee_id AS "Emp#"
m.last-name AS "Manager", m.employee_id AS "Mgr#"
from employees e left outer join employees m on e.manager_id = m.employee_id;

2. Modify question 1 to display all employees and their managers, even if the employee does not have a manager. Order the list alphabetically by the last name of the employee.

Select e.last-name AS "Employee", e.employee_id AS "Emp#"
m.last-name AS "Manager", m.employee_id AS "Mgr#"
from employees e left outer join employees m on e.manager_id = m.employee_id
order by "Employee";

3. Display the names and hire dates for all employees who were hired before their managers, along with their managers' names and hire dates. Label the columns Employee, Emp Hired, Manager, and Mgr Hired, respectively.

Select e.last-name AS "Employee", e.hire_date AS "Emp Hired",
m.last-name AS "Manager", m.hire_date AS "Mgr Hired"
from employees e left outer join employees m on e.manager_id = m.employee_id
where e.hire_date < m.hire_date;

4. Write a report that shows the hierarchy for Lex De Haan's department. Include last name, salary, and department id in the report.

Select last_name, salary, department_id from employees start
with employee_id = (select employee_id from employees where
last_name = 'De Haan') connect by prior employee_id = manager_id,

5. What is wrong in the following statement:

```
SELECT last_name, department_id, salary  
FROM employees  
START WITH last_name = 'King'  
CONNECT BY PRIOR manager_id = employee_id;
```

Select last-name, department_id, salary from employees start with last-name = 'King' connect by prior manager-id = employee-id;

6. Create a report that shows the organization chart for the entire employee table. Write the report so that each level will indent each employee 2 spaces. Since Oracle Application Express cannot display the spaces in front of the column, use - (minus) instead.

```
Select LPad('-', 2 * (level-1), '-') || last_name as "organization  
chart", salary, department_id  
from employees start with manager_id is NULL  
connect by prior employee_id = manager_id;
```

7. Re-write the report from 6 to exclude De Haan and all the people working for him.

```
Select LPad('-', 2 * (Level-1), '-') || last_name as "organization  
chart", salary, department_id  
from employees where last_name != 'De Haan' start with  
manager_id is NULL connect by prior employee_id = manager_id  
and last_name != 'De Haan';
```

Oracle Equijoin and Cartesian Product

1. Create a Cartesian product that displays the columns in the d_play_list_items and the d_track_listings in the DJs on Demand database.

Select * from d_play_list_items, d_track_listings;

2. Correct the Cartesian product produced in question 1 by creating an equijoin using a common column.

Select * from d_play_list_items i Join d_track_listings t
ON i.song-id = t.song-id;

3. Write a query to display the title, type, description, and artist from the DJs on Demand database.

Select s.title, t.type-description, t.type-description, s.artist
from d_songs s Join d_types t ON s.type-code = t.type-code;

4. Rewrite the query in question 3 to select only those titles with an ID of 47 or 48.

Select s.title, t.type-description, s.artist from d_songs
s join d_types t on s.type-code = t.type-code where
s.id IN(47, 48) .

5. Write a query that extracts information from three tables in the DJs on Demand database, the d_clients table, the d_events table, and the d_job_assignments table.

Select * from d_clients c Join d_events e on
c.client-number = e.client-number Join d_job_assignments
j on e.id = j.event-id; ✓

Group Functions

1. Define and give an example of the seven group functions: AVG, COUNT, MAX, MIN, STDDEV, SUM, and VARIANCE.

Select avg(salary) from employees;

Select count(*) from employees;

Select max(salary) from employees;

Select min(hire-date) from employees;

Select ~~mean~~ std dev(salary) from employees;

Select sum(salary) from employees;

Select variance(salary) from employees;

2. Create a query that will show the average cost of the DJs on Demand events. Round to two decimal places.

Select Avg(salary) From f_staffs where manager_id=19;

3. Find the average salary for Global Fast Foods staff members whose manager ID is 19.

Select sum(salary) from f_staffs where staff_id ~~in~~ in (12, 9);

4. Find the sum of the salaries for Global Fast Foods staff members whose IDs are 12 and 9.

Using the Oracle database, select the lowest salary, the most recent hire date, the last name of the person who is at the top of an alphabetical list of employees, and the last name of the person who is at the bottom of an alphabetical list of employees. Select only employees who are in departments 50 or 60

Select min(salary) As "Lowest salary",
max(hire_date) as "Most Recent Hire Date",
min(last_name) as "First Name Alphabetically",
max(last_name) as "Last Name Alphabetically",
from employees where department_id ~~in~~ in (50, 60);

5. Your new Internet business has had a good year financially. You have had 1,289 orders this year. Your customer order table has a column named total_sales. If you submit the following query, how many rows will be returned?

SELECT sum(total_sales) FROM orders;

~~Select~~

ans: One row will be returned because the SUM() function is an aggregate function used without a group by clause, collapsing all rows into a single summary row.

6. You were asked to create a report of the average salaries for all employees in each division of the company. Some employees in your company are paid hourly instead of by salary. When you ran the report, it seemed as though the averages were not what you expected—they were much higher than you thought! What could have been the cause?

The most likely cause is that the column used for salary (or the calculation) includes null values for hourly employees, and when calculating the avg function ignores nulls.

7. Employees of Global Fast Foods have birth dates of July 1, 1980, March 19, 1979, and March 30, 1969. If you select MIN(birthdate), which date will be returned?

The min() function on a date column returns the earliest date. The earliest date among the choices is march 30, 1969.

8. Create a query that will return the average order total for all Global Fast Foods orders from January 1, 2002, to December 21, 2002.

Select Avg(order_total) from f-orders where order_date
between '01-JAN-2002' And '21-DEC-2002';

9. What was the hire date of the last Oracle employee hired?

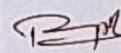
Select Max(hire_date) from employees;

10. Your new Internet business has had a good year financially. You have had 1,289 orders this year. Your customer order table has a column named total_sales. If you submit the following query, how many rows will be returned?

SELECT sum(total_sales)
FROM orders;

any : one row ~~one row~~



Evaluation Procedure	Marks awarded
Practice Evaluation (5)	5
Viva(5)	5
Total (10)	10
Faculty Signature	

Practice Problems -II

COUNT, DISTINCT, NVL

1. How many songs are listed in the DJs on Demand D_SONGS table?

Select count (*) from D_Songs;

2. In how many different location types has DJs on Demand had venues?

Select count (distinct loc_type) from D_Venues;

3. The d_track_listings table in the DJs on Demand database has a song_id column and a cd_number column. How many song IDs are in the table and how many different CD numbers are in the table?

Select count(song_id) As "Total Song IDS", count(DISTINCT cd_number)
as "Distinct CD number" from d-track_listings;

4. How many of the DJs on Demand customers have email addresses?

Select count(email) from d-Clients;

5. Some of the partners in DJs on Demand do not have authorized expense amounts (auth_expense_amt). How many partners do have this privilege?

Select count(auth_expense_amt) from D-partners;

6. What values will be returned when the statement below is issued?

ID	type	shoe_color
456	oxford	brown
463	sandal	tan

262	heel	black
433	slipper	tan

SELECT COUNT(shoe_color), → 4 (Total numbers of non-null colors)
 COUNT(DISTINCT shoe_color)

FROM shoes;

↳ 3 (The distinct colors are 'brown', 'tan', 'black')

7. Create a query that will convert any null values in the auth_expense_amt column on the DJs on Demand D_PARTNERS table to 100000 and find the average of the values in this column. Round the result to two decimal places.

Select Round(Avg(NVL(auth_expense_amt, 100000)), 2);
 from D_partners;

8. Which of the following statements is/are TRUE about the following query?

SELECT AVG(NVL(selling_bonus, 0.10))

FROM bonuses;

false a. The datatypes of the values in the NVL clause can be any datatype except date data.

True b. If the selling_bonus column has a null value, 0.10 will be substituted.

True c. There will be no null values in the selling_bonus column when the average is calculated.

false d. This statement will cause an error. There cannot be two functions in the SELECT statement.

4. Which of the following statements is true DQL about the following query?

SELECT color, size

FROM items;

A. Each color will appear only once in the results set.

B. Each size will appear only once in the results set.

C. Unique combinations of color and size will appear only once in the results set.

D. Each color and size combination will appear more than once in the results set.

Using GROUP BY and HAVING Clauses

1. In the SQL query shown below, which of the following are true about this query?
 - a. Kimberly Grant would not appear in the results set. *false*
 - b. The GROUP BY clause has an error because the manager_id is not listed in the SELECT clause. *false*
 - c. Only salaries greater than 16000 will be in the result set. *false*
 - d. Names beginning with K will appear after names beginning with D. *false*
 - e. Last names such as King and Foster will be returned even if they don't have salaries > 16000. *false*

```
SELECT last_name, MAX(salary)
FROM employees
WHERE last_name LIKE 'K%' GROUP BY
manager_id, last_name HAVING
MAX(salary) > 16000
ORDER BY last_name DESC.
```

2. Each of the following SQL queries has an error. Find the error and correct it. Use Oracle Application Express to verify that your corrections produce the desired results.

a. `SELECT manager_id
FROM employees
WHERE AVG(salary) < 16000
GROUP BY manager_id;`

Error: Avg(Salary) cannot be used in the where clause.

Correction: use the Having clause

b. `SELECT cd_number, COUNT(title)
FROM d_cds
WHERE cd_number < 93;`

- Error: A column (cd-number) and an aggregate function (Count(title)) are in the select list, but there is no grouping clause.*
- Correction: Add a group by clause for cd-number.*

c. `SELECT ID, MAX(ID), artist AS Artist FROM d_songs
WHERE duration IN('3 min', '6 min', '10 min')
HAVING ID < 50
GROUP BY ID;`

Error: aggregate functions & Having clause in redundant.

Correction: include artist in group by.

Summarizing Queries for practice

INSERT Statements

Students should execute DESC tablename before doing INSERT to view the data types for each column.
VARCHAR2 data-type entries need single quotation marks in the VALUES statement.

1. Give two examples of why it is important to be able to alter the data in a database.
 - Data correction: To fix errors in existing data
 - Business update: To reflect real world changes that impact the data
2. DJs on Demand just purchased four new CDs. Use an explicit INSERT statement to add each CD to the copy_d_cds table. After completing the entries, execute a SELECT * statement to verify your work.

CD_NUMBER	TITLE	PRODUCER	YEAR
97	Celebrate the Day	R&B Inc.	2003
98	Holiday Tunes for All Ages	Tunes are Us	2004
99	Party Music	Old Town Records	2004
100	Best of Rock and Roll	Old Town Records	2004

Insert into copy_d_cds values (97, 'Celebrate the day', 'R&B Inc', 2003);
Insert into copy_d_cds values (98, 'holiday tunes for all ages', 'Tunes are Us', 2004);
Insert into copy_d_cds values (99, 'party music', 'old town records', 2004);
Insert into copy_d_cds values (100, 'best of rock and roll', 'old town records', 2004);

3. DJs on Demand has two new events coming up. One event is a fall football party and the other event is a sixties theme party. The DJs on Demand clients requested the songs shown in the table for their events. Add these songs to the copy_d_songs table using an implicit INSERT statement.

Insert into copy_d_songs values (52, 'Surfing Summertime', 'not known', 12);
Insert into copy_d_song values (53, 'victory victory', '5 min', 12);

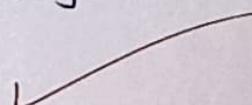
ID	TITLE	DURATION	TYPE_CODE
52	Surfing Summer	Not known	12
53	Victory Victory	5 min	12

4. Add the two new clients to the copy_d_clients table. Use either an implicit or an explicit INSERT.

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
6655	Ayako	Dahish	3608859030	dahisha@harbor.net
6689	Nick	Neuville	9048953049	nnicky@charter.net

Insert into copy_d_clients values (6655, 'Ayako', 'Dahish', '3608859030', 'dahisha@harbor.net');

Insert into copy_d_clients values (6689, 'Nick', 'Neuville', '9048953049', 'nnicky@charter.net');



5. Add the new client's events to the copy_d_events table. The cost of each event has not been determined at this date.

ID	NAME	EVENT_DATE	DESCRIPTION	COST	VENUE_ID	PACKAGE_CODE	THEME_CODE	CLIENT_NUMBER
110	Ayako Anniversary	07-Jul-2004	Party for 50, sixties dress, decorations	245	79		240	6655
115	Neuville Sports Banquet	09-Sep-2004	Barbecue at residence, college alumni, 100 people	315	87		340	6689

Insert into copy_d_events values (110, 'Ayako Anniversary', '07-Jul-2004', 'party for 50 sixties dress decoration, NULL, 245, 79, 240, 6655');

Insert into copy_d_events values (115, 'neuville sports banquet', '09-Sep-2004', 'barbecue at residence, college alumni, 100 people', NULL, 315, 87, 340, 6689);

6. Create a table called rep_email using the following statement:

```
CREATE TABLE rep_email (id NUMBER(3) CONSTRAINT rel_id_pk PRIMARY KEY, first_name
VARCHAR2(10), last_name VARCHAR2(10), email_address VARCHAR2(10))
```

Populate this table by running a query on the employees table that includes only those employees who are REP's.

Insert into rep_email (id, first_name, last_name, email_address)
 select employee_id, first_name, last_name, email from employees
 where job_id like '%REP';

Updating Column Values and Deleting Rows

NOTE: Copy tables in this section do not yet exist; students must create them.

If any change is not possible, give an explanation as to why it is not possible.

- Monique Tuttle, the manager of Global Fast Foods, sent a memo requesting an immediate change in prices. The price for a strawberry shake will be raised from \$3.59 to \$3.75, and the price for fries will increase to \$1.20. Make these changes to the copy_f_food_items table.

update copy-f-food-items set price = 3.75 where item-name = 'Strawberry Shake';

update copy-f-food-items set price = 1.20 where item-name = 'fries';

- Bob Miller and Sue Doe have been outstanding employees at Global Fast Foods. Management has decided to reward them by increasing their overtime pay. Bob Miller will receive an additional \$0.75 per hour and Sue Doe will receive an additional \$0.85 per hour. Update the copy_f_staffs table to show these new values. (Note: Bob Miller currently doesn't get overtime pay. What function do you need to use to convert a null value to 0?)

update copy-f-staffs set overtime-pay = NVL(overtime-pay, 0) + 0.75
where first-name = 'Bob' And last-name = 'Miller';

update copy-f-staffs set overtime-pay = NVL(overtime-pay, 0) + 0.85
where first-name = 'Sue' And last-name = 'Doe';

✓

3. Add the orders shown to the Global Fast Foods copy_f_orders table:

ORDER_NUMBER	ORDER_DATE	ORDER_TOTAL	CUST_ID	STAFF_ID
5680	June 12, 2004	159.78	145	9
5691	09-23-2004	145.98	225	12
5701	July 4, 2004	229.31	230	12

Copy-f-orders table: omitted for brevity, standard
INSERT statements.

4. Add the new customers shown below to the copy_f_customers table. You may already have added Katie Hernandez. Will you be able to add all these records successfully?

ID	FIRST_NAME	LAST_NAME	ADDRESS	CITY	STATE	ZIP	PHONE_NUMBER
145	Katie	Hernandez	92 Chico Way	Los Angeles	CA	98008	8586667641
225	Daniel	Spode	1923 Silverado	Denver	CO	80219	7193343523
230	Adam	Zurn	S Admiral Way	Seattle	WA		4258879009

you may not be able to add all records successfully.

Katie Hernandez might fail due to a primary key violation and Adam might fail due to a Not NULL constraint violation if last_name or zip are required columns.

5. Sue Doe has been an outstanding Global Foods staff member and has been given a salary raise. She will now be paid the same as Bob Miller. Update her record in copy_f_staffs.

update copy_f_staffs set salary = (select salary from copy_f_staffs where first_name = 'Bob' and last_name = 'Miller') where first_name = 'Sue' and last_name = 'Doe';

6. Global Fast Foods is expanding their staff. The manager, Monique Tuttle, has hired Kai Kim. Not all information is available at this time, but add the information shown at right.

ID	FIRST_NAME	LAST_NAME	BIRTHDATE	SALARY	STAFF_TYPE
25	Kai	Kim	3-Nov-1988	6.75	Order Taker

7. Now that all the information is available for Kai Kim, update his Global Fast Foods record to include the following: Kai will have the same manager as Sue Doe. He does not qualify for overtime. Leave the values for training, manager budget, and manager target as null.

update copy_g-staffs set manager_id = (Select manager_id from copy_g-staffs where first_name = 'sue' and last_name = 'doe'),
overtime_pay = NULL where first_name = 'Kai' and last_name = 'kim';

8. Execute the following SQL statement. Record your results.

DELETE from departments
WHERE department_id = 60;

- if no employees are in department 60 : 1 row deleted
- if ~~no~~ employees are in department 60 : error

9. Kim Kai has decided to go back to college and does not have the time to work and go to school.
Delete him from the Global Fast Foods staff. Verify that the change was made.

Delete from copy_g-staffs where first_name = 'Kai' and last_name = 'Kim';
Select * from copy_g-staffs where first_name = 'Kai' and last_name = 'Kim';

10. Create a copy of the employees table and call it lesson7_emp;

Once this table exists, write a correlated delete statement that will delete any employees from the lesson7_employees table that also exist in the job_history table.

Create Table lesson7_emp as select * from employees;
Delete from lesson7_emp where j.job_history_id exists (Select * from job_history; where j.employee_id = i.employee_id);

DEFAULT Values, MERGE, and Multi-Table Inserts

- When would you want a DEFAULT value?

You want a default value when most new records should automatically have specific, pre-defined values for a column, and you want to ensure the column is never left NULL without requiring the user to explicitly provide a value.

- Currently, the Global Foods F_PROMOTIONAL_MENU table START_DATE column does not have SYSDATE set as DEFAULT. Your manager has decided she would like to be able to set the starting date of promotions to the current day for some entries. This will require three steps:

- In your schema, Make a copy of the Global Foods F_PROMOTIONAL_MENU table using the following SQL statement.

Create table copy_f_promotional_menus as select * from F_promotional_menus;

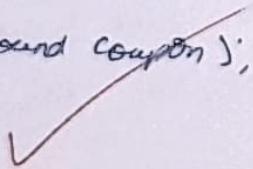
- Alter the current START_DATE column attributes using:

alter table copy_f_promotional_menus modify (start_date default sysdate);

- INSERT the new information and check to verify the results.

INSERT a new row into the copy_f_promotional_menus table for the manager's new promotion. The promotion code is 120. The name of the promotion is 'New Customer.' Enter DEFAULT for the start date and '01-Jun-2005' for the ending date. The giveaway is a 10% discount coupon. What was the correct syntax used?

Insert into copy_f_promotional_menus values(120, 'New Customer',
default, '01-Jun-2005', '10% discount coupon');



3. Alison Plumb, the event planning manager for CDs on Demand, has just given you the following list of CDs she acquired from a company going out of business. She wants a new updated list of CDs in inventory in an hour, but she doesn't want the original D_CDS table changed. Prepare an updated inventory list just for her.

- a. Assign new cd_numbers to each new CD acquired.

The user must manually assign new, unique numbers
eg: 201, 202, 203.

- b. Create a copy of the D_CDS table called manager_copy_d_cds. What was the correct syntax used?

create table manager_copy_d_cds as select * from D_CDS;

- c. INSERT into the manager_copy_d_cds table each new CD title using an INSERT statement. Make up one example or use this data:

Insert into manager_copy_d_cds values (20, 'Hello World Here I Am',
'middle earth Records', 1998);

20, 'Hello World Here I Am', 'Middle Earth Records', '1998' What was the correct syntax used?

- d. Use a merge statement to add to the manager_copy_d_cds table, the CDs from the original table. If there is a match, update the title and year. If not, insert the data from the original table. What was the correct syntax used?

merge into manager_copy_d_cds target using D_CDS source on
(target cd_number = source.cd_number) when matched then update set
target.title = source.title, target.year = source.year when not
matched then insert values (source.cd_number, source.title, source.year);

4. Run the following 3 statements to create 3 new tables for use in a Multi-table insert statement. All 3 tables should be empty on creation, hence the WHERE 1=2 condition in the WHERE clause.

```
CREATE TABLE sal_history (employee_id, hire_date, salary) AS  
SELECT employee_id, hire_date, salary  
FROM employees
```

WHERE 1=2;

```
CREATE TABLE mgr_history (employee_id, manager_id, salary)
```

```
AS SELECT employee_id, manager_id, salary
```

```
FROM employees
```

WHERE 1=2;

```
CREATE TABLE special_sal (employee_id, salary) AS
```

```
SELECT employee_id, salary
```

```
FROM employees
```

WHERE 1=2;

Once the tables exist in your account, write a Multi-Table insert statement to first select the employee_id, hire_date, salary, and manager_id of all employees. If the salary is more than 20000 insert the employee_id and salary into the special_sal table. Insert the details of employee_id, hire_date, and salary into the sal_history table. Insert the employee_id, manager_id, and salary into the mgr_history table.

You should get a message back saying 39 rows were inserted. Verify you get this message and verify you have the following number of rows in each table:

Sal_history: 19 rows

Mgr_history: 19 rows

Special_sal: 1

Insert all

when salary > 20000 then into special_sal (employee_id, salary)
values (employee_id, salary)

into sal_history values (employee_id, hire_date, salary)
into mgr_history values (employee_id, manager_id, salary)

Select employee_id, hire_date, salary, manager_id from employees;

Creating Tables

1. Complete the GRADUATE CANDIDATE table instance chart. Credits is a foreign-key column referencing the requirements table.

Complete the graduate candidate table instance chart with the syntax create the grad_candidatess table.

2. Write the syntax to create the grad_candidates table.

Create Table grad_candidates (candidate_id Number (6), first_name
VARCHAR2(30) NOT NULL, last_name VARCHAR2(30) NOT NULL, major
VARCHAR2(100), opp_date DATE, gpa Number (8,2)

3. Confirm creation of the table using DESCRIBE.

describe grad_candidates;

4. Create a new table using a subquery. Name the new table your last name - e.g., smith_table.
Using a subquery, copy grad_candidates into smith_table.

Create Table my_lastname_table as select * from grad_candidates;

5. Insert your personal data into the table created in question 4.

Insert your personal data into the table created in question 4.

Insert into grad_candidates values (1102, 'Kram', 'Kas'),
oppdate '01-Jan-2000', 63);

6. Query the data dictionary for each of the following:

- USER_TABLES returns information about all tables created by the current user.
- USER_OBJECTS returns information about all objects owned by the current user.
- USER_CATALOG or USER_CAT returns information about all tables, views and
cursors owned by the current user.

In separate sentences, summarize what each query will return.

Modifying a Table

Before beginning the practice exercises, execute a DESCRIBE for each of the following tables: o_employees and o_jobs. These tables will be used in the exercises. You will need to know which columns do not allow null values.

NOTE: If students have not already created the o_employees, o_departments, and o_jobs tables they should create them using the four steps outlined in the practice.

1. Create the three o_tables - jobs, employees, and departments - using the syntax:

```
Create table o-jobs (job_id varchar(10) primary key, job-title varchar(50) NOT NULL)
```

```
Create o-employees (employee_id number (primary key), first_name varchar(20), last_name varchar(25) NOT NULL, job_id varchar(10) references o-jobs (job_id), salary number);
```

2. Add the Human Resources job to the jobs table:

```
Insert into o-jobs values ('HR-REP', 'Human Resources Representative');
```

3. Add the three new employees to the employees table:

```
Insert into o-employees values (100, 'John', 'Doe', 'HR-REP', 50000);
```

```
Insert into o-employees values (101, 'Jane', 'Smith', 'IT-PRO', 60000);
```

4. Add Human Resources to the departments table:

```
Insert into o-departments values (90, 'Human Resources');
```

5. Why is it important to be able to modify a table?

- adapt to new data needs
- improve data integrity
- optimize performance

1. CREATE a table called Artists.

- a. Add the following to the table:

- artist ID
- first name
- last name
- band name
- email
- hourly rate
- song ID from d_songs table

- b. INSERT one artist from the d_songs table.

- c. INSERT one artist of your own choosing; leave song_id blank.

- d. Give an example how each of the following may be used on the table that you have created:

- 1) ALTER TABLE
- 2) DROP TABLE
- 3) RENAME TABLE
- 4) TRUNCATE
- 5) COMMENT ON TABLE

a. Explain to students how you want the DJs on Demand artist's table assignment to be completed. Students should be able to list the term followed by the SQL statement they used. For example:

- Create Table: `Create table table-name ;`
- Insert data: `Insert into table-name values ('...', ...);`
- Add column:
- Rename Table

2. In your o_employees table, enter a new column called "Termination." The datatype for the new column should be VARCHAR2. Set the DEFAULT for this column as SYSDATE to appear as character data in the format: February 20th, 2003.

```
alter Table o_employees add (
    Termination varchar(30) default To_Char(SYSDATE,
    'FMMonth DD, YYYY'));
```

3. Create a new column in the o_employees table called start_date. Use the TIMESTAMP WITH LOCAL TIME ZONE as the datatype.

```
alter table o_employees add (
    Start_date Timestamp with local time zone);
```

4. Truncate the o_jobs table. Then do a SELECT * statement. Are the columns still there? Is the data still there?

```
Truncate Table o_jobs;
Select * from o_jobs;
```

- c. Issue a select statement to see all rows and all columns from the copy_f_staffs table;

Select * from Copy-f-Staffs;

- d. For Sue Doe, update the salary to 2 and commit the change.

update Copy-f-Staffs set Salary=2 where first-name='Sue' and last-name='Doe';
commit;

- e. Issue a select statement to see all rows and all columns from the copy_f_staffs table;

Select * from Copy-f-Staffs;

- f. Now, issue a FLASHBACK QUERY statement against the copy_f_staffs table, so you can see all the changes made.

Select Salary, version_start_time, version_end_time, version_operation
from copy-f-Staffs version between timestampminvalue and maxvalue
where first-name='Sue' and last-name='Doe';

- g. Investigate the result of f), and find the original salary and update the copy_f_staffs table salary column for Sue Doe back to her original salary.

update Copy-f-Staffs set salary=5000 where first-name='Sue'
and last-name='Doe';
commit;

Evaluation Procedure	Marks awarded
Practice Evaluation (5)	5
Viva(5)	5
Total (10)	10
Faculty Signature	RPL