Assignment 1

Title: Creation, modification, deletion of table and insertion of data into table.

Objective: To understand how to create a table schema with data types and constraints. To learn insertion of data into a table.

The Assignment Covers the Course Outcome:	CO2
Bloom's Cognitive Domain:	Infer

Theory:

Create Table
Used for creation of a table schema.

```
Syntax: CREATE TABLE <table_name>(
          <column name><datatype>[(<size>)] [<constraint>],
          <column name><datatype>[(<size>)] [<constraint>],
          ...
          <column name><datatype>[(<size>)] [<constraint>],)
          );
```

Alter Table
Used for adding, modifying or deleting column(s) of a table,

Adding column(s)

```
Syntax:  ALTER TABLE <table_name>ADD (
          <column name><datatype>[(<size>)] [<constraint>],
          <column name><datatype>[(<size>)] [<constraint>],
           ...
          <column name><datatype>[(<size>)] [<constraint>],)
                                                          );
```

Modifying existing column(s)

```
Syntax: ALTER TABLE <table_name> MODIFY
          <column name><datatype>[(<size>)] [<constraint>],
          <column name><datatype>[(<size>)] [<constraint>],
          ...
          <column name><datatype>[(<size>)] [<constraint>];
```

Removing existing column

Syntax:ALTER TABLE <table_name>DROP COLUMN <column name>;

Insert into Table
Used for insertion of data into a table,

For all columns

Syntax: INSERT INTO <table_name>VALUES (<value>,<value>,.....,<value>)

For specific columns

Syntax:INSERT INTO <table_name>(<column name>, <column name>,.....,<column name>) VALUES (<value>,<value>,.....,<value>)

Problem Statements

1. Create the following tables and insert the data that follow. Specify necessary constraints while creating the tables.

EMP

Column name	Data type	Description
EMPNO	Number	Employee number
ENAME	Varchar	Employee name
JOB	Char	Designation
MGR	Number	Manager's Emp. Number
HIREDATE	Date	Date of joining
SAL	Number	Basic Salary
COMM	Number	Commission
DEPTNO	Number	Department Number

DEPT

Column name	Data type	Description
DEPTNO	Number	Department number
DNAME	Varchar	Department name
LOC	Varchar	Location of department

Data for EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	Smith	Clerk	7902	17/12/80	800		20
7499	Allen	Salesman	7698	20/2/81	1600	300	30
7521	Ward	Salesman	7698	22/2/81	1250	500	30
7566	Jones	Manager	7839	2/4/81	2975		20
7654	Martin	Salesman	7698	28/9/81	1250	1400	30
7698	Blake	Manager	7839	1/5/81	2850		30
7782	Clark	Manager	7839	9/6/81	2450		10
7788	Scott	Analyst	7566	9/12/82	3000		20
7839	King	President		17/11/81	5000		10
7844	Turner	Salesman	7698	8/9/81	1500	0	30
7876	Adams	Clerk	7788	12/1/83	1100		20

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7900	James	Clerk	7698	3/12/81	950		30
7902	Ford	Analyst	7566	4/12/81	3000		20
7934	Miller	Clerk	7782	23/1/82	1300		10

Data for DEPT table

DEPTNO	DNAME	LOC
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

2. Write SQL for the following:

   a) Display all the records in the *EMP* table.

   b) Display the details of all the employees who are working as managers.

   c) Display the details of the employee wjhose employee no. 7369.

   d) Display details of all the employees who joined on 1st May,1981.

   e) Display details of all the employees with salary greater than Rs.1500/-

   f) Display details of all theemployees who are not getting any commission.

   g) Display details of all the employees whose names are starting with "A".

   h) Add a new attribute Phone No. to the *EMP* table.

   i) Change the data type of attribute Job from char to varchar2.

   j) Remove the attribute Phone No. from the table*EMP*.

Discussions:

We have understood how to create a table schema with datatypes and constraints and insertion of data into table.

Questionnaire:

1. Which table should you create first? – explain.

2. Explain if any constraint is required to be specifird for the attribute *JOB*.

Ans:

(1) I should create DEPT table first because the EMP table has a foreign key (DEPTNO) referencing it. The referenced table must exist before defining a foreign key constraint to ensure referential integrity otherwise Oracle SQL will raise an error.

(2) No specific constraint is mandatory for the JOB (Varchar2 column) attribute in the EMP table. However, depending on business rules, we may apply constraints like:

- NOT NULL if every employee must have a job title.
- CHECK to restrict values (e.g. only predefined job roles)

Teacher's signature with date ....[illegible] 19/2/25....

Assignment 2

Title: Data storage using integrity constraints in SQL.

Objective: To understand how to work with constraints.

Assignment Covers the Course Outcome: CO2

Bloom's Cognitive Domain: Inten

Theory:

Primary Key

```
Syntax: CREATE TABLE <table_name>(
        <column name><datatype> [(<size>)] [<constraint>] constraint_name PRIMARY KEY, ...);
```

Foreign Key

```
Syntax:   CREATE TABLE <table_name>(
          <column name><datatype>,
          <column name><datatype>,
          ...
          [<CONSTRAINT>] constraint_name
          FOREIGN KEY (column1, column2, ... column_n)
                           REFERENCES <parent_table> (column1, column2, ... column_n));
```

Check

```
Syntax:CREATE TABLE <table_name>(
          <column name><datatype>,
          <column name><datatype>,
          ...

          [<CONSTRAINT>] constraint_name
          CHECK (<column name> condition));
```

Not Null

```
Syntax:CREATE TABLE <table_name>(
          <column name><datatype> NOT NULL,
          ( <column name><datatype> NOT NULL);
```

Problem Statements

1. Create the following tables:

Table Name: **Client_master**

Column Name	Data Type	Size	Attributes
Client_no	Varchar2	6	Primary key / 1st letter must start with 'C'
Name	Varchar2	20	Not null
City	Varchar2	15	
Pincode	Number	8	
State	Varchar2	15	
Bal_due	Number	10,2	

Table Name: **product_master**

Column Name	Data Type	Size	Attributes
Product_no	Varchar2	6	Primary key / 1st letter must start with 'P'
Description	Varchar2	15	Not null
Profit_percent	Varchar2 Number	4,2	Not null
Unit_measure	Varchar2	10	Not null
Qty_on_hand	Number	8	Not null
Reorder_lvl	Number	8	Not null
Sell_price	Number	8,2	Not null, cannot be 0
Cost_price	Number	8,2	Not null, cannot be 0

Table Name: **salesman_master**

Column Name	Data Type	Size	Attributes
Salesman_no	Varchar2	6	Primary key / 1st letter must start with 'S'
Salesman_name	Varchar2	20	Not null
Address1	Varchar2	10	Not null
Address1	Varchar2	10	
City	Varchar2	20	
Pincode	Number	7	
State	Varchar2	20	
Sal_amt	Number	8,2	Not null, cannot be 0
Tgt_to_get	Number	6,2	Not null, cannot be 0
Ytd_sales	Number	6,2	Not null
Remarks	Varchar2	20	

Table name: **sales_order**

Column Name	Data Type	Size	Attributes
Order_no	Varchar2	6	Primary key / 1st letter must start with 'O'
Order_date	date		
Client_no	Varchar2	6	Foreign key references client_no. of client_master
Dely_address	Varchar2	25	
Salesman_no	Varchar2	6	Foreign key references salesman_no of salesman_master
Dely_type	Char	1	Delivary: part(P) / full (F) default 'F'
Billed_yn	Char	1	
Dely_date	Date		Cannot be less than order_date
Order_status	Varchar2	10	Values ('in process', 'fulfilled','backorder','cancelled'

Table Name: **sales_order_details**

Column Name	Data Type	Size	Attributes
Order_no	Varchar2	6	Primary_key / foreign key ref. Order_no of the
Product_no	Varchar2	6	Primary_key / foreign key ref. Product_no of the
Qty_ordered	Number	8	
Qty_Disp	Number	8	
Product_rate	Number	10,2	

2. Insert the following data into their respective tables:

Data for CLIENT MASTER table:

ClientNo	Name	City	Pincode	State	BalDue
C00001	Ivan Bayross	Mumbai	400054	Maharashtra	15000
C00002	Mamta Mazumdar	Madras	780001	Tamil Nadu	0
C00003	Chhaya Bankar	Mumbai	400057	Maharashtra	5000
C00004	Ashwini Joshi	Bangalore	560001	Karnataka	0
C00005	Hansel Colaco	Mumbai	400060	Maharashtra	2000
C00006	Deepak Sharma	Mangalore	560050	Karanataka	0

Data for PRODUCT MASTER table:

ProductNo	Description	Profit Percent	Unit Measure	QtyOn Hand	ReorderLvl	SellPrice	CostPrice
P00001	T-Shirts	5	Piece	200	50	350	250
P0345	Shirts	6	Piece	150	50	500	350
P06734	Cotton Jeans	5	Piece	100	20	600	450
P07865	Jeans	5	Piece	100	20	750	500
P07868	Trousers	2	Piece	150	50	850	550
P07885	PuM-Overs	2.5	Piece	80	30	700	450
P07965	Denim Shirts	4	Piece	100	40	350	250
P07975	Lyers Tops	5	Piece	70	30	300	175
P08865	Skirts	5	Piece	75	30	450	300

Data for SALESMAN MASTER table:

SalesmanNo	Name	Address1	Address2	City	PinCode	State
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra
S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra
S00003	Raj	P-7	Bandra	Mumbai	400032	Maharashtra
S00004	Ashish	A/5	Juhu	Mumbai	400044	Maharashtra

SalesmanNo	SalAmt	TgtToGet	YtdSales	Remarks
S00001	3000	100	50	Good
S00002	3000	200	100	Good
S00003	3000	200	100	Good
S00004	3500	200	150	Good

Data for SALES ORDER table:

OrderNo	ClientNo	Order DelyDate	SalesmanNo	DelyType	BillYN	Dely OrderDate	OrderStatus
O19001	C00001	20-July-02	S00001	F	N	12-June-04	In Process
O19002	C00002	27-June-02	S00002	P	N	25-June-04	Cancelled
O46865	C00003	20-Feb-02	S00003	F	Y	18-Feb-04	Fulfilled
O19003	C00001	07-Apr-02	S00001	F	Y	03-Apr-04	Fulfilled
O46866	C00004	22-May-02	S00002	P	N	20-May-04	Cancelled
O19008	C00005	26-July-02	S00004	F	N	24-May-04	In Process

Data for SALES ORDER DETAILS table:

OrderNo	ProductNo	QtyOrdered	QtyDisp	ProductRate	
O19001	P00001	4	4	525	
O19001	P07965	2	1	8400	
O19001	P07885	2	1	5250	
O19002	P00001	10	0	525	
O46865	P07868	3	3	3150	
O46865	P07885	3	1	5250	
O46865	P00001	10	10	525	
O46865	P0345	4	4	1050	
→ O19003	P03453	2	2	1050	→ P03453 not present in product_master table
O19003	P06734	1	1	12000	
O46866	P07965	1	0	8400	
O46866	P07975	1	0	1050	
O19008	P00001	10	5	525	
O19008	P07975	5	3	1050	

3. Write SQL for the following:

a) Find the names of all clients having 'a' as the second letter in their names.
b) Find out the clients who stay in a city whose second letter is 'a'.
c) Find the list of all clients who stay in 'Bombay' or 'Delhi'
d) Print the list of clients whose bal_due is greater than value 10000.
e) Print the information from sales_order table for orders placed in the month of January.
f) Display the order information for client_no 'C00001' and 'C00002'.

g) Find products whose selling price is greater than 2000 and less than or equal to 5000.
h) Find products whose selling price is more than 1500. Calculate a new selling price as original selling price * .15. Rename the new column in the above query as new_price.
i) List the names, city and state of clients who are not in the state of 'Maharashtra'.
j) Count the total number of orders.
k) Calculate the average price of all the products.
l) Determine the maximum and minimum product prices. Rename the output as max_price and min_price respectively.
m) Count the number of products having price greater than or equal to 1500.
n) Find all the products whose qty_on_hand is less than reorder level.
o) Display the order number and day on which clients placed their order.

Discussions:

We have been familiarized with different types of constraints and their usage in writing SQL queries and creating tables.

Questionnaire:

1. Explain the significances of the different typesof constraints. What will be the problem if the constraints are violated?

Ans: Constraints in oracle SQL ensure data integrity and consistency.
Different types of constraints —
(i) NOT NULL prevents NULL values in a column. Violation causes an error when inserting /updating NULL

P.T.O.

Teacher's signature with date ......[illegible] 12/3/25......

(ii) UNIQUE ensures all values in a ~~ta~~ column are distinct. Violation occurs on duplicate entries.

(iii) PRIMARY KEY uniquely identifies rows (NOT NULL + UNIQUE). Violation occurs on duplicates or NULL values.

(iv) FOREIGN KEY ensures referential integrity by linking to a primary key in another table. Violation occurs when inserting an unmatched value or deleting a referenced row.

(v) CHECK enforces specific conditions on column values. Violation occurs if the condition is not met.

(vi) DEFAULT assigns a default value if none is provided. Violation occurs if an invalid default is used

Problem if the constraints are violated:

If constraints are violated, Oracle prevents the operation and throws an error message, ensuring the database maintains integrity and reliability. constraints help to avoid duplicate, inconsistent, or invalid data that could cause logical errors in applications.

Assignment 3

Title: Data retrieval from database using different functions in SQL.

Objective: To understand how to work with functions.

Assignment Covers the Course Outcome: CO2	
Bloom's Cognitive Domain: Infer	

Theory:

DUAL Table

The **DUAL table** is a special one-row, one-column **table** present by default in **Oracle** and other database installations. In **Oracle**, the **table** has a single VARCHAR2(1) column called DUMMY that has a value of 'X'. It is suitable for use in selecting a pseudo column such as SYSDATE, USER, etc.

Examples:

```
desc dual;
select sysdate from dual;
```

Aggregate Functions

MIN: To find the minimum value of an attribute.
Syntax: SELECT **MIN**(<aggregate_expression>)FROM <table name>   [WHERE <conditions>];

MAX: To find the maximum value of an attribute.
Syntax: SELECT **MAX**(<aggregate_expression>)FROM <table name>[WHERE <conditions>];

SUM:To find the sum of the values of an attribute.
Syntax: SELECT **SUM**(<aggregate_expression>)FROM <table name>[WHERE <conditions>];

AVG:To find the average of the values of an attribute.
Syntax: SELECT **AVG**(<aggregate_expression>)FROM <table name>[WHERE <conditions>];

String Functions

Function	Example	Result	Purpose
ASCII	ASCII(‘A’)	65	Returns an ASCII code value of a character.
CHR	CHR(’65’)	‘A’	Converts a numeric value to its corresponding ASCII character.
CONCAT	CONCAT(‘A’,’BC’)	‘ABC’	Concatenate two strings and return the combined string.
CONVERT	CONVERT(‘Ä Ê Í’, ‘US7ASCII’, ‘WE8ISO8859P1’)	‘A E I’	Convert a character string from one character set to another.
DUMP	DUMP(‘A’)	Typ=96 Len=1: 65	Return a string value (VARCHAR2) that includes the datatype code, length measured in bytes, and internal representation of a specified expression.
INITCAP	INITCAP(‘hi there’)	‘Hi There’	Converts the first character in each word in a specified string to uppercase and the rest to lowercase.
INSTR	INSTR(‘This is a playlist’, ‘is’)	3	Search for a substring and return the location of the substring in a string
LENGTH	LENGTH(‘ABC’)	3	Return the number of characters (or length) of a specified string
LOWER	LOWER(‘Abc’)	‘abc’	Return a string with all characters converted to lowercase.
LPAD	LPAD(‘ABC’,5,’*’)	‘**ABC’	Return a string that is left-padded with the specified characters to a certain length.
LTRIM	LTRIM(‘ ABC ’)	‘ABC ’	Remove spaces or other specified characters in a set from the left end of a string.

Function	Example	Result	Purpose
REPLACE	REPLACE('JACK AND JOND','J','BL');	'BLACK AND BLOND'	Replace all occurrences of a substring by another substring in a string.
RPAD	RPAD('ABC',5,'*')	'ABC**'	Return a string that is right-padded with the specified characters to a certain length.
RTRIM	RTRIM(' ABC ')	' ABC'	Remove all spaces or specified character in a set from the right end of a string.
SOUNDEX	SOUNDEX('sea')	'S000'	Return a phonetic representation of a specified string.
SUBSTR	SUBSTR('Oracle Substring', 1, 6)	'Oracle'	Extract a substring from a string.
TRANSLATE	TRANSLATE('12345', '143', 'bx')	'b2x5'	Replace all occurrences of characters by other characters in a string.
TRIM	TRIM(' ABC ')	'ABC'	Remove the space character or other specified characters either from the start or end of a string.
UPPER	UPPER('Abc')	'ABC'	Convert all characters in a specified string to uppercase.

Date Functions

Function	Purpose
Date	Returns a date string or the current date.
Date2Date	Converts one date format to a new format and returns the result.
DateAdd	Adds days, months, and years to the date and returns the result.
DateCnv	Converts a date specified with a two-digit

Function	Purpose
	year into a date containing a four-digit year value.
Day	Returns the day of the month number from a date and returns the result.
DayName	Returns the specified day name.
DaysInMonth	Returns the number of days in the specified month and year.
DaysInYear	Returns the number of days in the specified year.
DiffDate	Calculates the difference between two dates and returns a positive or negative value based on which date is earlier.
DiffDays	Returns the difference in days between two dates.
DiffMonths	Returns the difference in months between two dates.
DiffYears	Returns the difference in years between two dates.
LeapYear	Returns one (1) if the specified year is a leap year and zero (0) if it is not a leap year.
Month	Returns the month number from a date.
MonthName	Returns the specified month name.
WeekDay	Returns the week day number from a date.
Year	Returns the year from a date.

Problem Statements

1. Perform the following queries using DUAL.
   a) Display the current DATE and TIME.
   b) Multiply 2 by 2 .
   c) Find the absolute value of -15
   d) Calculate the square root of 5.
   e) Round off 15.19 to one decimal point:
   f) Display the name " IVAN BAYROSS" in LOWERCASE.
   g) Display the name " IVAN BAYROSS" in UPPERCASE.
   h) Add 5 months to the present date and print the output.
   i) Display the number of months between '02-JAN-01' and '02-JUL-01'
   j) Print the system date in the particular format 'DD/MMIYYYY'

2. Create the following table with the given constraints and insert 10 rows in the table.

. EMP (EMPNO , ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPT _NAME)
   a) EMPNO must be between 7000 and 8000
   b) ENAME must not exceed 10 characters
   c) MGR is managers EMPNO
   d) COMM (commission) must be under 1500 and defaults to O. Only who works as salesman gets certain commission.
   e) DEPT _NAME is the name of the department in which the employees works.

3. Write necessary SQL queries for the following.

   f) List the names of employees whose names have " i " as the second character .
   g) List names of employees who are not managers.
   h) Display the highest, lowest, Sum and average of all employees. Label the columns as "Maximum" "Minimum" "Total" and "Average".
   i) Modify-the above query to display the highest, lowest, sum and average salary for each job type.
   j) Determine the number of managers. Label the column "Number of managers"
   k) Find the employees who were hired after '01-jan-1980'
   l) Display the name of employee who earns maximum salary whose job is salesman
   m) Display the name of employee who earns minimum salary and whose job is clerk.
   n) Display the name of the department in which 'FORD' works.
   o) Display the name of the department whose salary is maximum.
   p) List the flame of the employee whose salary is more than 'TURNER'.

Assignment 4

Title: Data retrieval from database using JOIN in SQL - I

Objective: To understand how to work with JOIN.

Assignment Covers the Course Outcome: CO 2

Bloom's Cognitive Domain: Infer

Theory:

Natural join - A Natural joinis a join operation that creates an implicit join clause based on the common columns in the two tables being joined. Common columns are columns that have the same name in both tables.

Inner Join -The most frequently used and important of the joins is the inner join. They are also referred to as an EQUI JOIN. The inner joincreates a new result table by combining column values of two tables (table1 and table2) based upon the join-predicate.

Outer Join – In an Outer Join, the joined table retains each row—even if no other matching row exists. Outer joins subdivide further into left outer joins, right outer joins, and full outer joins, depending on which table's rows are retained (left, right, or both).

Consider the following tables

Employee

Empid	Deptid
3415	10
2241	20
3401	30
2202	40

Dept

Deptid	DeptName	Manager
10	Finance	George
20	Sales	Harriet
30	Production	Charle
40	Admin	David

Syntax of natural join for the above tables.

```
SELECT*FROMemployeeNATURALJOINdept
SELECT*FROMemployee,deptwhereemployee.Deptid=dept.Deptid ;
```

Problem Statements

1. Consider the following tables. Create the tables and insert sufficient records. Execute the queries that follow.

SAILORS(s_id , s_name , rating , age)
BOATS(b_id , b_name , color)
RESERVES(s_id , b_id , day)

i. s_id , b_id are primary keys of the tables SAILORS and BOATS.
ii. s_id , b_id together of the table RESERVES form the composite primary key.
iii. s_id , b_id are also the foreign keys references SAILORS and BOATS respectively.

a) Find the color of boats reserved by '**Tarun**'.
b) Find the sailor_id's and sailor_names who have reserved boats on '**Monday**'.
c) List boat_id's and boat names for '**red**' and '**green**' colors only.
d) Delete all the sailors information whose age is greater than 60.

2. Consider the following tables. Create the tables and insert sufficient records. Execute the queries that follow.

i. Teacher (Tid , Name , Dept)
ii. Subject (Subno , Subtitle)
iii. TaughtBy (Tid , Subno)
iv. Student (Rollno , Sname , City)

a) Get the names of all the teachers of 'Physics' department who teach 'Thermodynamics'.
b) Rename the subject 'DBMS' to 'RDBMS'.
c) Find out all the students who stay in 'Kolkata' and whose roll number is between 20 and 25.
d) Display all the students' information in descending order of their roll number who stay in 'Kolkata'.

Discussions:

We have been familiarized with several types of JOINS in Oracle SQL and data retrieval using several conditional joins for several tables.

Questionnaire:

1. Explain the difference between natural join and conditional join with examples.

Ans.

Feature	Natural Join	Conditional Join
Definition	Joins tables based on common column names and data types automatically	Joins tables based on a specified condition using the ON clause
Condition Requirement	Implicit (uses common column names.)	Explicit (requires condition using ON)
Duplicate Columns	Eliminates duplicate columns in the result.	Retains both columns unless explicitly selected.
Flexibility	Less flexible as it relies on common column names.	More flexible as it allows custom join conditions.
Example	SELECT * FROM EMPLOYEE NATURAL JOIN DEPARTMENT;	SELECT * FROM EMPLOYEE INNER JOIN DEPARTMENT ON EMPLOYEE.DEPTID = DEPARTMENT.DEPTID;

Teacher's signature with date ......[illegible] 19/3/25......

Assignment 5

Title: Data retrieval from database using JOIN in SQL - II

Objective: To understand how to work with JOIN.

Assignment Covers the Course Outcome: CO2	
Bloom's Cognitive Domain: Infers	

Theory:

Syntax for joins using tables Employee and Dept in Assignment 4

Equi-join

```
SELECT*FROMEMPLOYEEJOINdept
ONEMPLOYEE.Deptid=DEPT.Deptid;
```

Left Outer Join

```
SELECT*FROMemployee
LEFTOUTERJOINdeptONemployee.Deptid=dept.deptid;
```

Right Outer Join

```
SELECT*FROMemployeeRIGHTOUTERJOINdept
ONemployee.deptid=dept.deptid;
```

Full Outer Join

```
SELECT*FROMemployeeFULLOUTERJOINdepartment
ONemployee.deptid=dept.deptid;
```

Problem Statements

The mail-order database consists of the relations defined in the six schemes shown below:

EMPLOYEES (ENO , ENAME , ZIP , HDATE)
PARTS (PNO , PNAME , QOH , PRICE , LEVEL)
CUSTOMERS (CNO,CNAME,STREET,ZIP,PHONE)
ORDERS(ONO,CNO,ENO,PRECEIVED,SHIPPED)
ODETAILS(ONO , PNO , QTY)
ZIPCODES(ZIP , CITY)

Where

•The EMPLOYEES relation contains information about the employees of the company. The NO attribute is the primary key. The ZIP attribute is a foreign key referring to the ZIP-CODES table.

•The PARTS relation keeps a record of the inventory of the company. The record for each part includes its number and name as well as the quantity on hand, unit price and the re-order level. PNO is the primary key for the relation.

•The CUSTOMERS relation contains information about the customers of the mail-order company. Each customer is assigned a customer number, CNO, which serves as the primary key. The ZIP attribute is a foreign key referring to the ZIPCODES relation.

•The ORDERS relation contains information about the orders placed by customers, the employee who took the order, and the dates the order was received and shipped. ONO is the primary key. The CNO attribute is a foreign key referring to the CUSTOMERS relation, and the ENO attribute is a foreign key referring to the EMPLOYEES table.

•The ODETAILS relation contains information about the various parts ordered by the customers within a particular order. The combination of the ONO and PNO attributes forms the primary key. The ONO attribute is a foreign key referring to the ORDERS relation, and the PNO attribute is a foreign key referring to the PARTS relation.

•The ZIPCODES relation maintains information about the zip codes for various cities.ZIP is the primary key.

1. Create the above database using and execute the following queries:

a) Get PNO & PNAME values of parts that are priced less than 20.
b) Get PNO values for parts for which orders have been placed.
c) Get all the details of customers whose names begin with the letter "S".
d) Get the ONO & CNAME values for customers whose orders have not yet been shipped.
e) Get CNAME & ENAME pairs such the customer with name CNAME has placed and ordered through the employees with name ENAME.
f) Get the name of employees who was hired on the earliest date.

g) Retrieve the part number, part name and price of parts with price greater than 20000 in an ascending order of part number.

h) For each part, get PNO & PNAME values along with total sales.

i) Get the total quantity of parts 10601 that has been ordered.

j) Get the ENO values of employees from city "Mumbai".

Discussions:

· Implemented several types joins present in Oracle SQL like Inner Join, Natural Join, Left outer join, full outer join and Right outer join

Questionnaire:

1. Explain the different types of outer joins with examples.

Ans. There are 3 types of outer joins —

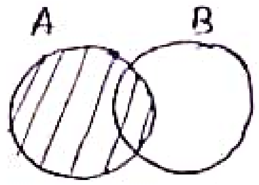
(1) Left Outer Join:

A Left outer Join returns all records from the left table and the matched records from the right table. If no match is found, NULL values are returned for the right table's columns.

Teacher's signature with date ....$\frac{15}{07/04/25}$..........

Syntax:

```
SELECT A.*, B.*
FROM TableA A
LEFT JOIN TableB B
ON A.id = B.id;
```



Customers (Left table)

CustomerID	Name
1	Alice
2	Bob
3	Charlie

Orders (Right table)

OrderID	CustomerID	Product
101	1	Laptop
102	2	Phone

Left Join Result:

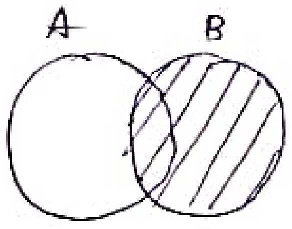
CustomerID	Name	OrderID	Product
1	Alice	101	Laptop
2	Bob	102	Phone
3	Charlie	NULL	NULL

(ii) Right Outer Join:

A Right outer Join returns all records from the right table and the matched records from the left table. If no match is found, NULL values are returned for the left table's columns.

Syntax:

```
SELECT A.*, B.*
FROM TableA A
RIGHT JOIN TableB B
ON A.id = B.id;
```



Right Join Result:

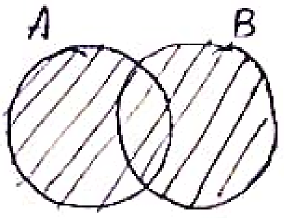
CustomerID	Name	OrderID	Product
1	Alice	101	Laptop
2	Bob	102	Phone
NULL	NULL	103	Tablet

(iii) Full Outer Join:

A full outer join returns all records from both tables. If there no match, NULL is returned in the missing side.

Syntax:

```
SELECT A.*, B.*
FROM TableA A
FULL JOIN TableB B
ON A.id = B.id;
```



Full Join result:

CustomerID	Name	OrderID	Product
1	Alice	101	laptop
2	Bob	102	Phone
3	Charlie	NULL	NULL
NULL	NULL	103	Tablet

Assignment 6

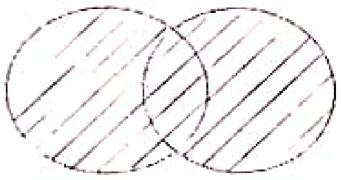
Title: Data retrieval from database using SET OPERATIONS in SQL.

Objective: To understand how to work with **SET OPERATIONS**.

Assignment Covers the Course Outcome: CO2	
Bloom's Cognitive Domain: Infer	

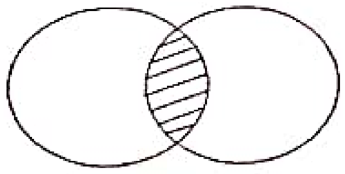
Theory:

Union - UNION is used to combine the results of two or more Select statements. However it will eliminate duplicate rows from its result set. In case of union, number of columns and data type must be same in both the tables.

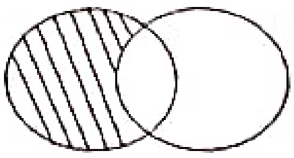


Union All - This operation is similar to Union. But it also shows the duplicate rows.

Intersect – Intersect operation is used to combine two SELECT statements, but it only returns the records which are common from both SELECT statements. In case of Intersect the number of columns and data type must be same.



Minus- Minus operation combines result of two select statements and return only those result which belongs to first set of result.



Syntax for the following examples:

Table: First

ID	Name
1	Abhi
2	adam

Table: Second

ID	Name
2	adam
3	chester

UNION

select * from first UNION select * from second

Output:

ID	NAME
1	abhi
2	adam
3	Chester

UNION ALL

select * from first UNION ALL select * from second

OUTPUT

ID	NAME
1	abhi
2	adam
2	adam
3	Chester

INTERSECT

select * from first INTERSECT select * from second

OUTPUT

ID	NAME
2	adam

MINUS

select * from First MINUS select * from second

OUTPUT

ID	NAME
1	abhi

Problem Statements

1. Create the following tables and execute the queries that follow:

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futter	Maria Anders	Obere Str.57	Berlin	12209	Germany
2	Ana helados	Ana Trujillo	Avda. Construction 2222	Mexico D.F	05021	Mexico
3	Antonio Moreno	Antonio Moreno	Mataderos 2312	Mexico D.F	05023	Mexico

Suppliers

SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country
1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	Berlin	12209	Germany
2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	Mexico D.F.	05023	Mexico
3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA

a) Selects all the different cities (only distinct values) from the "Customers" and the "Suppliers" tables.
b) Select all cities from the "Customers" and "Suppliers" tables.
c) Select all German cities from the "Customers" and "Suppliers" tables.
d) Select all Customer name and supplier name from "Customers" tables where city name is common in both.
e) Select all country names from Supplier Table which don't have any customer in customer table from its own country.

Assignment 7

Title: Data retrieval using subqueries

Objective: To retrieve data using single row and multiple row subquery.

Assignment Covers the Course Outcome: CO3	
Bloom's Cognitive Domain: Create	

Theory:

Examples of Sub Query:

Consider the table "employee" with attributes First_Name, Last_name, Department_ID, Salary and "department" with attributes Department_ID, Department_Name, Location_ID.

Single Row Sub Query:

```
SELECT first_name, salary, department_id
FROM employees WHERE salary = (SELECT MIN (salary) FROM employees);
```

Multiple row sub query:

```
SELECT first_name, department_id
FROM employees
WHERE department_id IN (SELECT department_idFROM departments WHERE LOCATION_ID =
100)
```

Problem Statements

1. Use the EMP table given in your workspace execute the following queries.
   a) Display the name of employee who earns maximum salary.
   b) Display the name of employee who earns maximum salary whose job is salesman.
   c) Display the name of employee who earns minimum salary and whose job is clerk.
   d) Display the department whose average salary is maximum.
   e) List the name of the employee whose salary is more than 'TURNER'
   f) List the name of employee who joined after ALLEN
   g) Display the name of the department in which 'FORD' works.
   h) Display the name of the department whose salary is maximum.

i) Display the name of the city(location) in which 'SMITH' works.
j) Display the name of the city in which the manager works.
k) Display the grade of the employee named 'MARTIN'
l) List the employees who earns more than every employee in 'DALLAS'
m) Display the name of the department which has no employee.
n) List the name of the employee who joined in the same date of 'ADAMS'
o) List the name of the department who gets commission.
p) List the employees who earn the lowest salary in their respective department.

Discussions:

Retrieving data using single row and multiple row subqueries.

Questionnaire:

1. Explain the use of *EXISTS, ALL, SOME, ANY* in subqueries.
2. Give an example where the same output is produced by using i) JOIN and ii) SUBQUERY.
3. Explain SELF JOIN with an example.

P.T.O.

Teacher's signature with date .... [illegible] 5/4/15

(1) Explain the use of EXISTS, ALL, SOME, ANY in subqueries.

Ans:

(i) EXISTS:

- Checks if the subquery returns any row.
- Returns TRUE if at least one row exists in the subquery result.

Use case: when we want to test for existence.

SQL:
```
SELECT ENAME FROM EMP E
WHERE EXISTS (SELECT 1 FROM DEPT D WHERE
              D.DEPTNO = E.DEPTNO);
```
↓
Returns employees who are assigned to a valid department.

(ii) ALL:

- Compares a value to all values returned by a subquery.
- The condition must be TRUE for all rows.

SQL:
```
SELECT ENAME FROM EMP WHERE SAL > ALL (SELECT
SAL FROM EMP WHERE DEPTNO = 30);
```
↓
Returns employees whose salary is higher than every employees in department 30.

(iii) SOME/ANY:

- Both are synonyms in Oracle
- Compares a value to any one value returned by the subquery.
- The condition is true if it matches at least one value.

SQL:
```
SELECT ENAME FROM EMP WHERE SAL > ANY (SELECT
SAL FROM EMP WHERE DEPTNO = 20);
```
↓
Returns employees who earn more than at least one employee in department 20.

(2) Give an example where the same output is produced by using i) JOIN and ii) SUBQUERY

Ans: Displaying the names of employees and the names of their departments.

Using JOIN:

```
SELECT E.ENAME, D.DNAME FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO;
```

Using Subquery:

```
SELECT ENAME, (SELECT DNAME FROM DEPT WHERE
 DEPT.DEPTNO = E.DEPTNO) DNAME FROM EMP E;
```

(3) Explain Self Join with an example.

Ans: A self join is a regular join where a table is joined with itself. It is useful when ~~yo~~ we want to compare rows within the same table.

Example: Find all employees along with their manager's name from the EMP table.

SQL:
```
SELECT E.ENAME EMPLOYEE, M.ENAME MANAGER
FROM EMP E
JOIN EMP M ON E.MGR = M.EMPNO;
```

Result:

```
EMPLOYEE    MANAGER
SMITH       FORD
  :           :
JONES       KING
```

Assignment 8

Title: Working with views and DML commands

Objective: To retrieve data by creating views and manipulate data by DML.

Assignment Covers the Course Outcome:	Create
Bloom's Cognitive Domain:	CO4

Theory:

View creation from Single table:

Consider the CUSTOMERS table having the following attributes:
ID, NAME, AGE, ADDRESS, SALARY

To create a view from CUSTOMERS table with customer name and age:

```
SQL > CREATE VIEW CUSTOMERS_VIEW AS
SELECT name, age
FROM  CUSTOMERS;
```

Now, you can query CUSTOMERS_VIEW in similar way as you query an actual table as follows.

```
SQL > SELECT * FROM CUSTOMERS_VIEW ;
```

With CHECK Option:

```
SQL> CREATE VIEW CUSTOMERS_VIEW AS
SELECT name, age
FROM CUSTOMERS
WHERE age IS NOT NULL
WITH CHECK OPTION;
```

Consider the following two tables:

The attributes of EMP table are EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO.
The attributes of DEPT table are DEPTNO, DNAME, LOC.

The following statement creates the emp_dept view:

```
CREATE VIEW emp_dept AS SELECT emp.empno, emp.ename, emp.deptno, emp.sal, dept.dname,
dept.loc FROM emp, dept WHERE emp.deptno = dept.deptno AND dept.loc IN ('DALLAS', 'NEW
YORK', 'BOSTON');
```

Data Manipulation Language (DML) commands

UPDATE
```
UPDATE emp_deptSET sal = sal * 1.10 WHERE deptno = 10;
```

DELETE
```
DELETE FROM emp_deptWHERE ename = 'SMITH';
```

INSERT
```
INSERT INTO emp_dept (ename, empno, deptno)
VALUES ('KURODA', 9010, 40);
```

DROPPING Views
```
DROP VIEW emp_dept;
```

Problem Statements

1. Create the following tables.

SALESPEOPLE (SNUM, SNAME, CITY, COMM) CUSTOMER (CNUM, CNAME, CITY, RATING, SNUM) ORDERS (ONUM, AMT, ODATE, CNUM, SNUM)

2. Create views that show

a) All of the customers who have highest rating
b) The number of salesperson in each city
c) The average and total orders for each salesperson
d) Each salesperson with multiple customers

3. Create a view Salespeople_Customer _Orders containing the following columns:

Salespeople_Customer _ Orders (SalespeopleN arne, CustomerName, OrderNumber)

4. Create a view of SALESPEOPLE table considering only two fields, i.e. SNUM and COMM. Thorough this view someone can insert or modify commission values between Rs. 1000 and Rs. 2000.

Assignment 9

Title: Programming Using PL/SQL.

Objective:

- To know the basics for writing program in PL/SQL.
- To know the basics of embedding SQL.

Assignment Covers the Course Outcome: CO6

Bloom's Cognitive Domain: Work

Theory:

PL/SQL is a block-structured language, meaning that PL/SQL programs are divided and written in logical blocks of code. Each block consists of three sub-parts:

```
DECLARE
<declarations section>
BEGIN
<executable command(s)>
EXCEPTION
<exception handling>
END;
```

Declaration of Data Types:

```
DECLARE
num1 INTEGER;
num2 REAL;
num3 DOUBLE PRECISION;
BEGIN
null;
END;
```

Conditional Control:

```
IF <condition> THEN
        <action>
ELSEIF <condition> THEN
        <action>

ELSE
     <action>
END IF;
```

Loop Structure:

```
WHILE <condition>
LOOP
        <action>
END LOOP;
```

Embedding SQL:

Consider the TABLE with attributes as COL1, COL2, and COL3.

```
SET SERVEROUTPUT ON
DECLARE
<variable1>[<datatype(size)>];
<variable2>[<datatype(size)>];
<variable3>[<datatype(size)>];
<variable4>[<datatype(size)>];
BEGIN
<variable2> := &<variable2>;
SELECT <COL3> INTO <VARIABLE4> FROM <TABLE> WHERE <COL2>=<VARIABLE2>;
IF <CONDITION> THEN
    <ACTION>
END IF;
END;
```

Problem Statements

2. Write the following in PL/SQLwith corresponding Outputs for the following:

a) Accept an account number if the account balance is less than the minimum balance then Rs. 100/- has to be deducted from the balance.
b) Find the 1st three characters of the employee name who were hired in the year 1981. Display the result with ename as nickname and the other columns too.
c) Find the employees from the emp table whose salary are among the top seven.
d) The office at Chicago has decided to give a 20 % bonus to all its salesman whose salaries are <=15000. Populate the bonus table with all such records.
e) For the Boston office, two employees were recruited one of grade2 and one of grade5 at the lowest remuneration possible. Insert 2 such records in this table.
f) There was a hike in salary of all employees of the company. For
    a. grade 1:- 15%,
    b. grade 2:- 12%,
    c. grade 3:- 10%,
    d. grade 4:- 10% and
    e. grade 5:- 5%.
g) Update the employees' records. Also store the old information and the date on which his/her salaries changed.

Assignment 10

Title: Programming Using Trigger.

Objective: To know the basics for writing program using trigger

Assignment Covers the Course Outcome: CO5	
Bloom's Cognitive Domain: Apply	

Theory:

Trigger Syntax

```
CREATE TRIGGER <TriggerName>
{BEFORE, AFTER}
{DELETE, INSERT, UPDATE} ON <TableName>
REFERENCING { OLD AS old, NEW AS new}
FOR EACH  ROW [ WHEN CONDITION ]
BEGIN
< SQL  /  PL/SQL subprogram body>
END ;
```

Where ,

BEFORE: Indicates that the Database engine fires the trigger BEFORE executing the triggering statement.

AFTER: Indicates that the Database engine fires the trigger AFTER executing the triggering statement.

DELETE: Indicates that the Database engine fires the trigger whenever a DELETE statement removes a row from the table.

INSERT: Indicates that the Database engine fires the trigger whenever an INSERT statement adds a row to the table.

UPDATE: Indicates that the Database engine fires the trigger whenever an UPDATE statement changes a value in one of the columns specified in the OF clause . If the OF clause is omitted , the database engine fires the trigger whenever an UPDATE statement changes a value in any column of the table.

ON: Specifiesthe schema and the name of the table, which the trigger is to be created.

REFERENCING: Specifies correlation names.Correlation names can be used in the SQL / PLSQL block and WHEN clause of a row trigger to refer specifically to old and new values of the current row.
The default correlation names are OLD & NEW.

FOR EACH ROW: Designates the trigger to be a row trigger. The database engine fires a row trigger once for each row that is affected by the triggering statement and meets the optional trigger constraint defined in the WHEN clause. If the value is omitted the trigger is a statement trigger.

WHEN: Specifies the trigger restriction.The trigger restriction contains a SQL condition that must be satisfied for the database engine to fire the trigger.

Problem Statement

1. Create a table "Employee1" with attributes EID, ENAME, CITY, DESIGNATION, SALARY, PERKS and insert data for at least three employees at first. Write the following using triggers.

a) Update the Salary column of Employee1 table before inserting any record in Employee1 table. The SALARY will be deducted by Rs.300/- when the PERKS exceeds Rs. 500/-.

b) Modify the salary of Employee table before updating the record of the EMPLOYEE table. If the SALARY is less than and equal to Rs. 500/-, then set it to Rs. 10,000/-. Otherwise, set salary to Rs. 15,000/-.

c) Create two tables "EMP" and "EMP_BACKUP" having same attributes. The attributes are empid, ename, salary. Write a program using trigger in such a way that whenever a row will be inserted in "EMP" table, a copy of the row will also be inserted in "EMP_BACKUP" at the same time.

Discussions:

Understanding the concept of Trigger in PL/SQL.

Questionnaire:

1. Compare row level and statement level triggers with suitable examples.

2. Discuss the difference between *before* and *after* triggers.

Ans (1)

Aspect	Row-level Triggers	Statement-level Triggers
Execution	Once per affected row	Once per SQL statement
Use case	Row-wise auditing or validation.	Logging or enforcing rules at table level.
OLD/NEW Access	Yes	NO
Example	Log each employee's salary change.	Log that an update occurred in employees

(2)

Aspect	BEFORE Triggers	AFTER Triggers
Execution	Runs before the DML operation.	Runs after the DML operation.
Use case	To validate or modify data before insert/update	To log or take action after data changes.

Teacher's signature with date 2/5/25