**Labs**

**Lab 1**

CREATE TABLE SUPPLIERS

(

    SID CHAR(4),

    SNAME VARCHAR2(40),

    STATUS NUMERIC(3,0),

    SCITY VARCHAR2(30),

    CONSTRAINT SUPPLIER\_PK PRIMARY KEY (SID)

);

CREATE TABLE PARTS

(

    PID CHAR(4),

    PNAME VARCHAR2(40),

    COLOR VARCHAR2(20),

    WEIGHT INTEGER,

    PCITY VARCHAR2(30),

    CONSTRAINT PARTS\_PK PRIMARY KEY (PID)

);

CREATE TABLE SHIPMENTS

(

    SPID CHAR(8),

    SID CHAR(4),

    PID CHAR(4),

    QUANTITY NUMBER(6,0),

    CONSTRAINT SHIPMENTS\_PK PRIMARY KEY (SPID, PID, SID),

    CONSTRAINT SHIPMENTS\_FK\_SUPPLIERS FOREIGN KEY (SID) REFERENCES SUPPLIERS,

    CONSTRAINT SHIPMENTS\_FK\_PARTS FOREIGN KEY (PID) REFERENCES PARTS

);

SELECT \*

FROM PARTS;

SELECT \*

FROM SHIPMENTS;

ALTER TABLE SUPPLIERS

ADD (Sadness VARCHAR2(300)

);

ALTER TABLE SUPPLIERS

MODIFY (SID VARCHAR2(4)

);

ALTER TABLE SUPPLIERS

MODIFY (SNAME VARCHAR2(50) NOT NULL

);

UPDATE SUPPLIERS SET SCITY= 'CHICAGO' where SID='S7';

UPDATE SUPPLIERS SET Sadness= '1000 Fremont Ave 220' where SID='S5';

**Lab 2**

SELECT \*

FROM SH.CHANNELS;

SELECT COUNT(\*)

FROM SH.SALES;

---Display the first and last names of all customers.

SELECT \*

FROM SH.CUSTOMERS;

SELECT CUST\_FIRST\_NAME, CUST\_LAST\_NAME

FROM SH.CUSTOMERS;

---Display  the  names  (&  the  city)  of  customers  living  in  Barcelona?  Report the FULL NAME INSTEAD OF FIRST AND LAST NAME.

SELECT CUST\_FIRST\_NAME ||' '|| CUST\_LAST\_NAME as "Full Name", CUST\_CITY

FROM SH.CUSTOMERS

WHERE cust\_city = 'Barcelona';

---Display names of customers with credit limit less than $5000. Print their first name, last name, gender, year of birth, income level, and credit limit

SELECT CUST\_FIRST\_NAME, CUST\_LAST\_NAME, cust\_gender, cust\_year\_of\_birth, cust\_income\_level, cust\_credit\_limit

FROM SH.CUSTOMERS

WHERE cust\_credit\_limit < 5000;

---Display  the  promotions  that  started  in  March  1999.  Print  promo\_id, promotion name, promotion category, and promotion begin date?

SELECT \*

FROM SH.PROMOTIONS;

SELECT PROMO\_ID, PROMO\_NAME, PROMO\_CATEGORY, PROMO\_BEGIN\_DATE

FROM SH.PROMOTIONS

WHERE PROMO\_BEGIN\_DATE >= '1-March-1999' AND PROMO\_BEGIN\_DATE <= '31-March-1999'

ORDER BY (PROMO\_BEGIN\_DATE);

    ---or

SELECT \*

FROM SH.PROMOTIONS

WHERE TO\_CHAR (PROMO\_BEGIN\_DATE, 'YYYY-MM-DD') >= '1999-03-01' AND

TO\_CHAR (PROMO\_BEGIN\_DATE, 'YYYY-MM-DD') <= '1999-03-31';

---(a and b) Formulate the above query, using BETWEEN function. Sort the output by the promotion begin date in Ascending or Descending order.

SELECT PROMO\_ID, PROMO\_NAME, PROMO\_CATEGORY, PROMO\_BEGIN\_DATE

FROM SH.PROMOTIONS

WHERE PROMO\_BEGIN\_DATE >= '1-March-1999' AND PROMO\_BEGIN\_DATE <= '31-March-1999'

ORDER BY (PROMO\_BEGIN\_DATE) DESC;

---List the products in product category ‘Electronics’ in alphabetical order of their names. Print all columns.

SELECT \*

FROM SH.PRODUCTS;

SELECT \*

FROM SH.PRODUCTS

WHERE PROD\_CATEGORY = 'Electronics'

ORDER By PROD\_NAME;

---List the products in product subcategory ‘Home Audio’. Print all columns.

SELECT \*

FROM SH.PRODUCTS

WHERE PROD\_SUBCATEGORY = 'Home Audio';

---List the product categories present in the PRODUCTS table?

SELECT DISTINCT PROD\_CATEGORY

FROM SH.PRODUCTS;

SELECT COUNT (DISTINCT PROD\_CATEGORY)

FROM SH.PRODUCTS;

--- How many sales transactions have sales amount greater than $1700?

SELECT COUNT (PROD\_ID)

FROM SH.SALES

WHERE AMOUNT\_SOLD > 1700;

Lab 3

--- Display the number of product subcategories in each product category.

SELECT PROD\_CATEGORY, COUNT (DISTINCT PROD\_SUBCATEGORY) AS "NUMBER OF subcategories"

FROM SH.PRODUCTS

GROUP BY PROD\_CATEGORY;

--- (a) Display the number of products in each product category which have a list price greater than $500.

SELECT PROD\_CATEGORY, COUNT (DISTINCT PROD\_ID) AS "NO OF PRODUCTS"

FROM SH.PRODUCTS

WHERE PROD\_LIST\_PRICE > 500

GROUP BY PROD\_CATEGORY;

--- (b) Display the number of products in each product category which have a list price greater than $500. Display only those product categories which have at least 2 products.

SELECT PROD\_CATEGORY, COUNT (DISTINCT PROD\_ID) AS "NO OF PRODUCTS"

FROM SH.PRODUCTS

WHERE PROD\_LIST\_PRICE > 500

GROUP BY PROD\_CATEGORY

HAVING COUNT(DISTINCT PROD\_ID) >= 2;

--- Print the total amount of sales in dollars and total quantity sold per customer. Round the results into 2 decimals.

SELECT CUST\_ID, ROUND(SUM(AMOUNT\_SOLD),2) AS "TOTAL AMOUNT SOLD IN DOLLARS", ROUND(SUM(QUANTITY\_SOLD),2) AS "TOTAL QUANTITY\_SOLD"

FROM SH.SALES

GROUP BY CUST\_ID;

SELECT CUST\_ID, ROUND(SUM(AMOUNT\_SOLD),2) AS "TOTAL AMOUNT SOLD IN DOLLARS", ROUND(SUM(QUANTITY\_SOLD),2) AS "TOTAL QUANTITY\_SOLD"

FROM SH.SALES

GROUP BY CUST\_ID

ORDER BY ROUND(SUM(AMOUNT\_SOLD),2) DESC;

--- Print the range of total amount of sales in dollars and range of total quantity sold per customer. Round all numbers to 2 decimals.

**Lab 4**

- What is the total\_sales in dollars in each country. Order the results by the descending order of sales.

SELECT co.COUNTRY\_NAME, ROUND(SUM(s.AMOUNT\_SOLD)) as "Total SALES $"

FROM sh.SALES s

JOIN sh.CUSTOMERS cu ON(s.CUST\_ID = cu.CUST\_ID)

JOIN sh.COUNTRIES co ON(cu.COUNTRY\_ID = co.COUNTRY\_ID)

GROUP BY co.COUNTRY\_NAME

ORDER BY SUM(s.AMOUNT\_SOLD) DESC;

--- Display  the  no.  of  distinct  promotions  (use  promo\_id  to  count  the  distinct promotions) per country.

--- Sort the results by the descending order of no. of distinct promotions.

SELECT co.COUNTRY\_NAME, count(distinct p.PROMO\_ID) as "No. of distinct promotions"

FROM sh.PROMOTIONS p

JOIN sh.SALES s ON(p.PROMO\_ID = s.PROMO\_ID)

JOIN sh.CUSTOMERS cu ON(cu.CUST\_ID = s.CUST\_ID)

JOIN sh.COUNTRIES co ON(cu.COUNTRY\_ID = co.COUNTRY\_ID)

GROUP BY co.COUNTRY\_NAME

ORDER BY count(distinct p.PROMO\_ID) DESC;

--- Display the products sold in Brazil in the fiscal year 1999. Print the product

--- name only. Sort the results by product name.

SELECT \*

FROM sh.PRODUCTS;

**Lab 5**

- Write a query to identify the channel that received maximum sales (in dollars) in fiscalyear 2000.

SELECT c.channel\_desc, ROUND(SUM(s.AMOUNT\_SOLD),0) as "Total Sales"

FROM SH.SAlES s JOIN SH.TIMES t ON (s.time\_id = t.time\_id)

    JOIN SH.CHANNELS c ON (c.channel\_id = s.channel\_id)

    WHERE t.fiscal\_year = 2000

    GROUP BY c.channel\_desc

    ORDER BY ROUND(SUM(s.AMOUNT\_SOLD),0) DESC

    FETCH NEXT 1 ROW ONLY;

SELECT c.channel\_desc

FROM SH.SAlES s JOIN SH.TIMES t ON (s.time\_id = t.time\_id)

    JOIN SH.CHANNELS c ON (c.channel\_id = s.channel\_id)

    WHERE t.fiscal\_year = 2000

    GROUP BY c.channel\_desc

    ORDER BY ROUND(SUM(s.AMOUNT\_SOLD),0) DESC

    FETCH NEXT 1 ROW ONLY;

--- Display  the  customer  Id,  first  name,  last  name  and  year  of  birth  of  customers  born

--- between years 1960 and 1970, who live in the city of Barcelona, and belong to the group

--- of customers with the maximum income level.

SELECT c1.CUST\_ID, c1.CUST\_FIRST\_NAME, c1.Cust\_last\_name, c1.cust\_year\_of\_birth

FROM SH.customers c1

WHERE (c1.CUST\_YEAR\_OF\_BIRTH between 1960 and 1970) AND

(c1.cust\_city = 'Barcelona') AND

cust\_income\_level = (

    SELECT MAX(c2.cust\_income\_level)

FROM SH.customers c2

);

--- Return the number of customers that had made at least one purchase with the company.

    --- a. Try executing this query using a single SELECT loop.

    --- b. Try executing this query using the EXISTS function.

SELECT (COUNT(DISTINCT(c.CUST\_ID)))

FROM sh.customers c JOIN sh.sales s ON (c.cust\_id = s.cust\_id);

--- or

SELECT COUNT(DISTINCT(CUST\_ID)) as "Customers that have made purchases"

FROM sh.sales;

--- or

SELECT COUNT(c.CUST\_ID)

FROM sh.CUSTOMERS c

WHERE EXISTS

    (

        SELECT s.cust\_id

        FROM sh.sales s

        where s.cust\_id = c.CUST\_ID

    );

--- Find the number of customers who have not made a single purchase with the company

    --- a. Try executing this query using a single SELECT loop.

    --- b. Try executing this query using the NOT EXISTS function.

SELECT COUNT(c.CUST\_ID)

FROM sh.CUSTOMERS c

WHERE NOT EXISTS

    (

        SELECT s.cust\_id

        FROM sh.sales s

        where s.cust\_id = c.CUST\_ID

    );

--- or

SELECT COUNT(c.CUST\_ID)

FROM sh.CUSTOMERS c

WHERE c.cust\_id NOT IN

    (

        SELECT s.cust\_id

        FROM sh.sales s

    );

--- or

SELECT COUNT(c.CUST\_ID)

FROM sh.sales s RIGHT OUTER JOIN sh.CUSTOMERS c ON(s.cust\_id = c.CUST\_ID)

WHERE s.QUANTITY\_SOLD IS NULL;

--- Which  product  has  the  highest  average  promotion  cost.  Display  the  product  name,

--- product category, and its average promotion cost rounded to two decimals.

SELECT pr.PROD\_NAME, pr.PROD\_CATEGORY, AVG(prom.promo\_cost)

FROM sh.PRODUCTS pr JOIN sh.sales s ON (pr.prod\_id = s.PROD\_ID)

                    JOIN sh.promotions prom ON (prom.promo\_id = s.promo\_id)

GROUP BY pr.PROD\_NAME, pr.PROD\_CATEGORY

ORDER BY AVG(prom.promo\_cost) DESC

FETCH NEXT 1 ROWS ONLY;