Table

Create table sales(ID NUMBER PRIMARY KEY,CID NUMBER NOT NULL,SALEDATE DATE NOT NULL,TOTAL NUMBER NOT NULL);

Create table sales\_detail(ID NUMBER ,Pname varchar2(15) ,QUANTITY number, constraint pk primary key(id,pname));

CREATE TABLE VENDOR(ID NUMBER PRIMARY KEY,COMPANYNAME VARCHAR2(20) NOT NULL,PHONENO NUMBER(11) NOT NULL,SALESMAN VARCHAR2(15) NOT NULL,CITY VARCHAR2(10) NOT NULL );

CREATE TABLE INVOICE\_DETAILS(IID NUMBER, PID NUMBER,QUANTITY NUMBER, CONSTRAINT IPK PRIMARY KEY (IID,PID) );

create table bill(ID NUMBER,EBILL NUMBER NOT NULL,RBILL NUMBER NOT NULL,BDATE DATE NOT NULL,CONSTRAINT BPK PRIMARY KEY (ID,BDATE));

CREATE TABLE invoice (

ID NUMBER PRIMARY KEY,

VID NUMBER NOT NULL,

IDATE DATE NOT NULL,

TOTAL NUMBER NOT NULL,

CONSTRAINT FK FOREIGN KEY (VID) REFERENCES vendor(ID)

);

Sequences

1)CREATE SEQUENCE order\_s

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 1000

NOCACHE;

2)

CREATE SEQUENCE invoice\_s

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 1000

NOCACHE;

3)CREATE SEQUENCE bill\_s

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 1000

NOCACHE;

Triggers

1)

CREATE OR REPLACE TRIGGER update\_product\_category

AFTER UPDATE ON category

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

UPDATE product

SET categoryname = :NEW.name

WHERE categoryname = :OLD.name;

COMMIT;

END;

2)

CREATE OR REPLACE TRIGGER delete\_product\_category

BEFORE DELETE ON category

FOR EACH ROW

BEGIN

DELETE FROM product

WHERE categoryname = :OLD.name;

END;

/

Procedures + DCL

CREATE OR REPLACE PROCEDURE DeleteRecordByID (

p\_id IN NUMBER

) AS

BEGIN

DELETE FROM bill

WHERE id = p\_id;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

RAISE;

END;

/

->FUNCTION

CREATE OR REPLACE FUNCTION CalculateTotalExpenses(monthNumber IN NUMBER)

RETURN NUMBER

IS

totalExpenses1 NUMBER;

totalExpenses2 NUMBER;

totalExpenses3 NUMBER;

totalSum NUMBER;

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

SELECT SUM(sal) INTO totalExpenses1 FROM emp;

SELECT total INTO totalExpenses2 FROM (SELECT SUM(total) AS total FROM invoice WHERE EXTRACT(MONTH FROM idate) = monthNumber);

SELECT SUM(ebill + rbill) INTO totalExpenses3 FROM bill WHERE EXTRACT(MONTH FROM bdate) = monthNumber;

totalSum := totalExpenses1 + totalExpenses2 + totalExpenses3;

COMMIT; -- Commit the autonomous transaction

RETURN totalSum;

END;

/

CREATE OR REPLACE TRIGGER update\_product\_quantity

AFTER INSERT ON sales\_detail

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

v\_product\_name sales\_detail.pname%TYPE;

v\_quantity sales\_detail.quantity%TYPE;

BEGIN

-- Retrieve the product name and quantity from the inserted row

v\_product\_name := :NEW.pname;

v\_quantity := :NEW.quantity;

-- Update the quantity in the product table

UPDATE product

SET quantity = quantity - v\_quantity

WHERE name = v\_product\_name;

-- Commit the transaction

COMMIT;

END;

/

CREATE OR REPLACE TRIGGER add\_quantity\_to\_product

AFTER INSERT ON invoice\_details

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

v\_product\_name invoice\_details.pname%TYPE;

v\_quantity invoice\_details.quantity%TYPE;

BEGIN

-- Retrieve the product name and quantity from the inserted row

v\_product\_name := :NEW.pname;

v\_quantity := :NEW.quantity;

-- Update the quantity in the product table

UPDATE product

SET quantity = quantity + v\_quantity

WHERE name = v\_product\_name;

-- Commit the transaction

COMMIT;

END;

/

Joins

SELECT COUNT(DISTINCT p.category\_id) AS total\_categories

FROM product p

JOIN category c ON p.category\_id = c.category\_id;

SELECT SUM(Sales.amount) AS total\_sales

FROM Sales

JOIN Products ON Sales.product\_id = Products.product\_id

WHERE Products.category = 'your\_category';