

DATABASE TEST PART II SQL, CONCEPTION ANS SPSS

- 1) Using SPSS syntax output frequency table for salary using class intervals give highest salary=512 and lowest =17 and frequency histogram for the salary

Intervals =  $(512-17)/8=61.8$

Class1 = 17 - 78

Class2 = 79 – 140

Class3 = 141 – 202

Class4 = 203 – 264

Class5 = 265 – 326

Class6 = 327 – 388

Class7 = 389 – 450

Class8 = 451 – 512

**Answer**

```
DO IF (Salary >=17 AND Salary <=78)
```

```
    COMPUTER SalaryCat=Class1
```

```
ELSE IF (Salary >= 79 AND Salary<=140)
```

```
    COMPUTER SalaryCat=Class2
```

```
ELSE IF (Salary >=141 AND Salary <=202)
```

```
    COMPUTER SalaryCat=Class3
```

```
ELSE IF (Salary>=203 AND Salary<=264)
```

```
    COMPUTER SalaryCat=Class4
```

```
ELSE IF (Salary>=265 and Salary<=326)
```

```
    COMPUTER SalaryCat=Class5
```

```
ELSE IF (Salary >=327 AND Salary <=388)
```

```
    COMPUTER SalaryCat=Class6
```

```
ELSE IF (Salary>=389 AND Salary <=450)
```

```
    COMPUTER SalaryCat=Class7
```

```
ELSE
```

```
    COMPUTER SalaryCat=Class8
```

```
END IF
```

```
VARIABLES LABELS SalaryCat "Salary Categories"
```

```
VALUES LABELS Class1"17-78" 2 " 79-140" 3"141-202" 4 "203-264" 5"265-326" 6"327-382"  
7"389-450" 8"451-512"
```

```
FREQUENCIES SalaryCat
```

```
GRAPH
```

**2) Use part of SPSS data file to answer questions below:**

EmpNo	Name	Gender	Age	Edl	MST	Taste	Salary
793673	Carine	2	25	3	1	3	195
376700	Jeff	1	30	1	2	1	221
130230	Pascal	1	34	1	1	1	190
234423	Jean	1	23	3	2	1	80
676767	Benedetta	1	25	3	1	3	200

**Write SPSS syntax to output:**

**a) All females who are below university level;**

**Answer:**

```
DO IF (Gender = 2)
  SELECT IF (Edl<3)
END IF
LIST
```

**b) Output all females who are below university level and married;**

**Answer:**

```
DO IF (Gender = 2)
  SELECT IF (Edl <3 AND MST=1)
END IF
LIST
```

## TEST ANALYSE

### 3) Les règles de gestions :

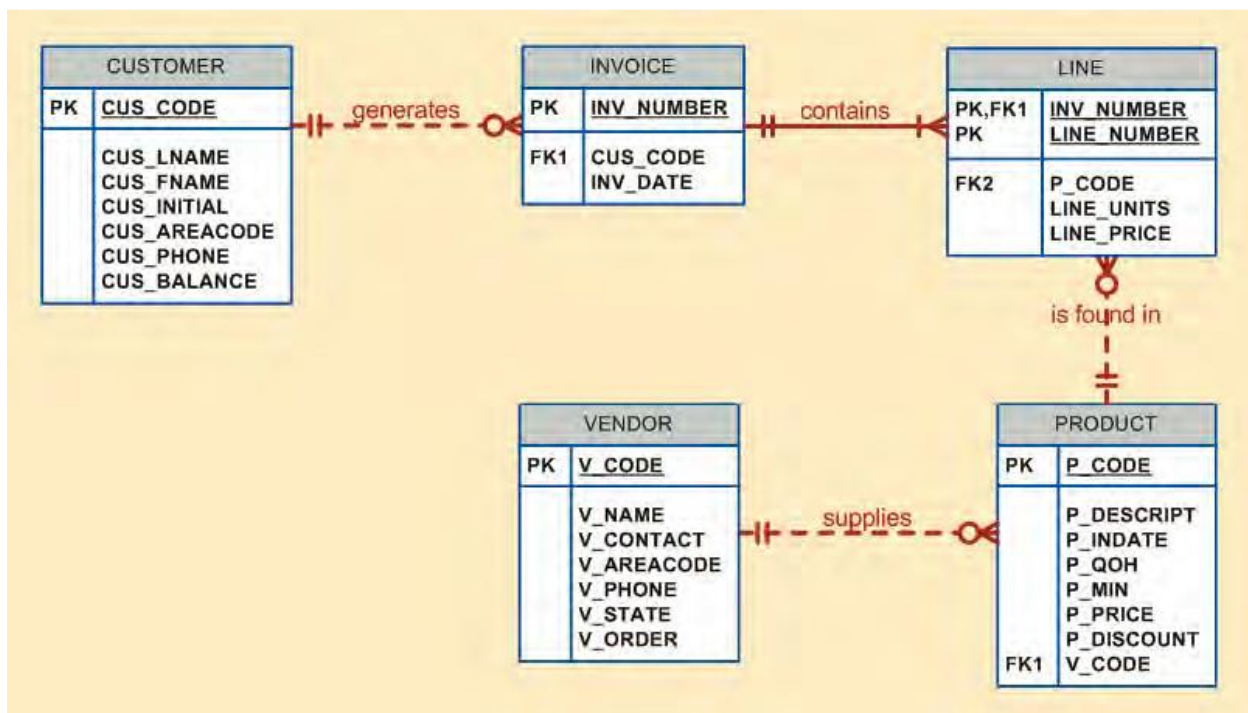
- ✓ Un client peut générer de nombreuses factures. Chaque facture est générée par un client.
- ✓ Une facture contient une ou plusieurs lignes de facture. Chaque ligne de facturation est associée à une seule facture.
- ✓ Chaque ligne de facturation fait référence à un produit. Un produit peut être trouvé dans beaucoup de lignes de facture. (Vous pouvez vendre plus d'un marteau à plus d'un client.)
- ✓ Un fournisseur peut fournir de nombreux produits. Certains fournisseurs ne fournissent pas (encore?) Des produits. (Par exemple, une liste de fournisseurs peut inclure des fournisseurs potentiels.)
- ✓ Si un produit est fourni par un fournisseur, ce produit est fourni par un seul fournisseur.
- ✓ Certains produits ne sont pas fournis par un fournisseur. (Par exemple, certains produits peuvent être produits en interne ou achetés sur le marché libre.)

QUESTIONS :

Avec les règles de gestions ci-dessus:

- Faire une représentation utilisant le model CHEN ;
- Faire une représentation utilisant le model Crwo's Foot ;

Answer :



- c) **Faire un Modèle conceptuel des données (MCD) ;**
- d) **Faire un modèle logique des données (MLD) ;**
- e) **Faire un modèle physique des données ;**

4) Dans le cadre de l'implémentation de la base des données conçue à la question précédente veuillez :

- a) avec une requête SQL de définition DDL, créer la base des données sous le nom "Ventes" et les tables ;

Réponse :

**CREATE TABLE FOUNISSEUR (**

V\_CODE INTEGER NOT NULL UNIQUE,  
 V\_NAME VARCHAR(35) NOT NULL,  
 V\_CONTACT VARCHAR(15) NOT NULL,  
 V\_AREACODE CHAR(3) NOT NULL,  
 V\_PHONE CHAR(8) NOT NULL,  
 V\_STATE CHAR(2) NOT NULL,  
 V\_ORDER CHAR(1) NOT NULL,  
 PRIMARY KEY (V\_CODE));

**CREATE TABLE PRODUIT (**

P\_CODE VARCHAR(10) NOT NULL UNIQUE,  
 P\_DESCRIPT VARCHAR(35) NOT NULL,  
 P\_INDATE DATE NOT NULL,  
 P\_QOH SMALLINT NOT NULL,  
 P\_MIN SMALLINT NOT NULL,  
 P\_PRICE NUMBER(8,2) NOT NULL,  
 P\_DISCOUNT NUMBER(5,2) NOT NULL,  
 V\_CODE INTEGER,  
 PRIMARY KEY (P\_CODE),  
 FOREIGN KEY (V\_CODE) REFERENCES VENDOR ON UPDATE CASCADE);

**CREATE TABLE CLIENT (**

CUS\_CODE NUMBER PRIMARY KEY,  
 CUS\_LNAME VARCHAR(15) NOT NULL,  
 CUS\_FNAME VARCHAR(15) NOT NULL,  
 CUS\_INITIAL CHAR(1),  
 CUS\_AREACODE CHAR(3) DEFAULT '615' NOT NULL  
 CHECK(CUS\_AREACODE IN ('615','713','931')),  
 CUS\_PHONE CHAR(8) NOT NULL,  
 CUS\_BALANCE NUMBER(9,2) DEFAULT 0.00,  
 CONSTRAINT CUS\_UI1 UNIQUE (CUS\_LNAME, CUS\_FNAME));

**CREATE TABLE FACTURE (**

INV\_NUMBER NUMBER PRIMARY KEY,  
 CUS\_CODE NUMBER NOT NULL REFERENCES CUSTOMER(CUS\_CODE),  
 INV\_DATE DATE DEFAULT SYSDATE NOT NULL,  
 CONSTRAINT INV\_CK1 CHECK (INV\_DATE > TO\_DATE('01-JAN-2010','DD-MON-YYYY')));

**CREATE TABLE LINE (**

INV\_NUMBER NUMBER NOT NULL,  
 LINE\_NUMBER NUMBER(2,0) NOT NULL,

```
P_CODE VARCHAR(10) NOT NULL,  
LINE_UNITS NUMBER(9,2) DEFAULT 0.00 NOT NULL,  
LINE_PRICE NUMBER(9,2) DEFAULT 0.00 NOT NULL,  
PRIMARY KEY (INV_NUMBER, LINE_NUMBER),  
FOREIGN KEY (INV_NUMBER) REFERENCES INVOICE ON DELETE CASCADE,  
FOREIGN KEY (P_CODE) REFERENCES PRODUCT(P_CODE),  
CONSTRAINT LINE_UI1 UNIQUE(INV_NUMBER, P_CODE));
```

**b) faire l'insertion des données dans la table PRODUIT avec une requête SQL:**

Answer :

```
INSERT INTO PRODUCT  
VALUES ('BRT-345','Titanium drill bit','18-Oct-09', 75, 10, 4.50, 0.06, NULL);
```

**c) Sélectionnez tous les produits livrés par les fournisseurs**

Answer :

```
SELECT *  
  
FROM PRODUIT INNER JOIN FOURNISSEUR ON PRODUIT.P_CODE=FOURNISSEUR.P_CODE;
```

**d) Mettre à jour la table PRODUIT ou le Produit code= '13-Q2/P2' avec une requête sql**

```
UPDATE PRODUCT  
SET P_INDATE = '18-JAN-2010', P_PRICE = 17.99, P_MIN = 10  
WHERE P_CODE = '13-Q2/P2';
```

**e) Sélectionnez Par exemple une liste pour tous les produits dont les prix sont Entre 50 \$ et 100 \$ avec une requête SQL**

```
SELECT *  
FROM PRODUCT  
WHERE P_PRICE BETWEEN 50.00 AND 100.00;
```

**f) Avec une sous requête SQL veuillez sélectionner les fournisseurs qui ont livrés les produit en utilisant l'opérateur IN;**

```
SELECT V_CODE, V_NAME  
FROM VENDOR  
WHERE V_CODE IN (SELECT V_CODE FROM PRODUCT);
```

**g) En utilisant l'Opérateur EXISTS SQL, répertoirer tous les fournisseurs, mais uniquement s'il existe des produits avec la quantité disponible, moins du double de la quantité minimale**

```
SELECT *  
FROM VENDOR  
WHERE EXISTS (SELECT * FROM PRODUCT WHERE P_QOH < P_MIN * 2);
```

**h) Supposer que vous voulez connaître tous les clients qui ont acheté des produits 13-Q2 / P2Et 23109-HB. Mais dans ce cas, vous voulez connaître tous les clients qui ont acheté les deux produits, pas seulement un**

**Answer :**

```
SELECT DISTINCT CUSTOMER.CUS_CODE, CUSTOMER.CUS_LNAME
FROM CUSTOMER,
(SELECT INVOICE.CUS_CODE FROM INVOICE NATURAL JOIN LINE
WHERE P_CODE = '13-Q2/P2') CP1,
(SELECT INVOICE.CUS_CODE FROM INVOICE NATURAL JOIN LINE
WHERE P_CODE = '23109-HB') CP2
WHERE CUSTOMER.CUS_CODE = CP1.CUS_CODE AND CP1.CUS_CODE = CP2.CUS_CODE;
```