

Assignment 4

5.14 (page 172)

Consider the following six relations for an order-processing database application in a company:

CUSTOMER(Cust#, Cname, City)

ORDER(Order#, Odate, Cust#, Ord_amt)

ORDER_ITEM(Order#, Item#, Qty)

ITEM(Item#, Unit_price)

SHIPMENT(Order#, Warehouse#, Ship_date)

WAREHOUSE(Warehouse#, City)

Here, **Ord_amt** refers to total dollar amount of an order; **Odate** is the date the order was placed; and **Ship_date** is the date an order (or part of an order) is shipped from the warehouse.

Assume that an order can be shipped from several warehouses.

Specify the foreign keys for this schema, stating any assumptions you make.

What other constraints can you think of for this database?

Foreign Keys (assumptions in parentheses):

Order# (Same domain in ORDER and ORDER_ITEM, a tuple in ORDER will reference a tuple in ORDER_ITEM),

Item# (Same domain in ORDER_ITEM and ITEM, a tuple in ORDER_ITEM will reference a tuple in ITEM),

Warehouse# (Same domain in SHIPMENT and WAREHOUSE, a tuple in SHIPMENT will reference a tuple in WAREHOUSE),

Cust# (Same domain in ORDER and CUSTOMER, a tuple in ORDER will reference a tuple in CUSTOMER).

Other constraints:

Semantic integrity constraints:

- "Ord_amt" and "Qty" can never be negative.

5.16 (page 173)

Consider the following relations for a database that keeps track of student enrollment in courses and the books adopted for each course:

STUDENT(Ssn, Name, Major, Bdate)

COURSE(Course#, Cname, Dept)

ENROLL(Ssn, Course#, Quarter, Grade)

BOOK_ADOPTION(Course#, Quarter, Book_isbn)

TEXT(Book_isbn, Book_title, Publisher, Author)

Specify the foreign keys for this schema, stating any assumptions you make.

Foreign Keys (assumptions in parentheses):

Ssn (Same domain in STUDENT and ENROLL, a tuple in STUDENT will reference a tuple in ENROLL),

Course# (Same domain in COURSE and ENROLL, a tuple in CUSTOMER will reference a tuple in ENROLL),

Quarter (Same domain in ENROLL and BOOK_ADOPTION, a tuple in ENROLL will reference a tuple in BOOK_ADOPTION),

Book_isbn (Same domain in BOOK_ADOPTION and TEXT, a tuple in BOOK_ADOPTION will reference a tuple in TEXT).

6.13. Write SQL update statements to do the following on the database schema shown in Figure 1.2.

- a. Insert a new student, <'Johnson', 25, 1, 'Math'>, in the database.
- b. Change the class of student 'Smith' to 2.
- c. Insert a new course, <'Knowledge Engineering', 'cs4390', 3, 'cs'>.
- d. Delete the record for the student whose name is 'Smith' and whose student number is 17.

a. **INSERT INTO** STUDENT
VALUES ('Johnson', 25, 1, 'MATH');

b. **UPDATE** STUDENT
SET Class = 2
WHERE Name = 'Smith';

c. **INSERT INTO** COURSE
VALUES ('Knowledge Engineering', 'cs4390', 3, 'cs');

d. **DELETE FROM** STUDENT
WHERE Name = 'Smith'
AND Student_number = 17;

6.14. Design a relational database schema for a database application of your choice.

- Declare your relations using the SQL DDL.
- Specify a number of queries in SQL that are needed by your database application.
- Based on your expected use of the database, choose some attributes that should have indexes specified on them.
- Implement your database, if you have a DBMS that supports SQL.

a.

```
CREATE TABLE    SOCCER_LEAGUE
(
    League_id      INT          NOT NULL,
    League_name    VARCHAR(50)  NOT NULL,
    PRIMARY KEY (League_id),
);

CREATE TABLE    TEAM
(
    Team_id        INT          NOT NULL,
    League_id      INT          NOT NULL,
    Country_id     INT          NOT NULL,
    Tname          VARCHAR(50)  NOT NULL,
    Wins           INT,
    Losses         INT,
    Draws         INT,
    PRIMARY KEY (Team_id),
    FOREIGN KEY (League_id) REFERENCES SOCCER_LEAGUE (League_id),
    FOREIGN KEY (Country_id) REFERENCES COUNTRY (Country_id),
);

CREATE TABLE    COUNTRY
(
    Country_id     INT          NOT NULL,
    Country_name   VARCHAR(50)  NOT NULL,
    PRIMARY KEY (Country_id),
);

CREATE TABLE    PLAYERS
(
    Player_id      INT          NOT NULL,
    League_id      INT          NOT NULL,
    Team_id        INT          NOT NULL,
    Country_id     INT          NOT NULL,
    Player_name    VARCHAR(50)  NOT NULL,
    Age           INT,
    Bdate          DATE,
    Position       VARCHAR(50),
    Jersey_number  INT,
    PRIMARY KEY (Player_id),
    FOREIGN KEY (League_id) REFERENCES SOCCER_LEAGUE (League_id),
    FOREIGN KEY (Team_id) REFERENCES TEAM (Team_id),
    FOREIGN KEY (Country_id) REFERENCES COUNTRY (Country_id),
);
```

b.

Retrieve the team_name, position, and jersey_number of the player(s) whose name is 'Lionel Messi'.

```
SELECT    Tname, Position, and Jersey_number
FROM      PLAYER
WHERE      Player_name = 'Lionel Messi';
```

Retrieve the name and country of all teams playing in the league 'Champion's League'.

```
SELECT    Tname, Country_name
FROM      TEAM, COUNTRY, SOCCER_LEAGUE
WHERE      League_name = 'Champion's League';
```

Retrieve all the attribute values of any TEAM who plays in the LEAGUE 'Champion's League'

```
SELECT    *
FROM      TEAM
WHERE      League_name = 'Champion's League';
```

c. The attributes **League_name**, **Team_name**, **Player_name**, and **Position** are some attributes that should have indexes on them.

d.