Chapter-6 Review

Yogesh Kalapala
yxk9640
1001879640

Q1
Sol:
Database schema: Database schema is a namespace used to specify the structure, relations, and constraints required for the whole database.
Database state: The data of a database at a particular instance of time is called database state.

A database state changes every time the data is added but the schema remains the same.
When an Instance is added the state changes but schema changes only when a data item is added.
The state is empty the first time DB is added to DBMS.

Example of Schema…

Schema at T=0

STUDENT

Name	Student_number	class

COURSE

Course_name	Course_number	Credit_hours	Department

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

GRADE_REPORT

Student_number	Section_identifier	Grade

Schema at T=1

STUDENT

Name	Student_number	class	Major

COURSE

Course_name	Course_number	Credit_hours	Department

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

GRADE_REPORT

Student_number	Section_identifier	Grade

PREREQUISITE

Course_number	Prerequisite_number

At State-1

STUDENT

Name	Student_number	Class
Sam	19	1
Blake	9	2

COURSE

Course_name	Course_number	Credit_hours	Department
Data Structures	CS5312	3	CS
Database	CS5302	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
201	CS5312	Fall	15	Ching
202	CS5302	Spring	15	Mark

GRADE_REPORT

Student_number	Section_identifier	Grade
19	201	A
9	202	A

PREREQUISITE

Course_number	Prerequisite_number
CS5312	CS3201
CS5302	CS3310

At State-2

At T2
STUDENT

Name	Student_number	Class
Sam	19	1
Blake	9	2
Clay	23	1

COURSE

Course_name	Course_number	Credit_hours	Department
Data Structures	CS5312	3	CS
Database	CS5302	3	CS
Discrete	MATH5312	4	MATH

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
201	CS5312	Fall	15	Ching
202	CS5302	Spring	15	Mark
301	CS5333	Spring	16	Bailey

GRADE_REPORT

Student_number	Section_identifier	Grade
19	201	A
9	202	A
23	301	A

PREREQUISITE

Course_number	Prerequisite_number
CS5312	CS3201
CS5302	CS3310

Yogesh Kalapala
yxk9640
1001879640

Chapter-6 Review

Q2
Sol:
Entity Integrity constraints: PRIMARY KEY, UNIQUE

Referential Integrity constraints: FOREIGN KEY.

Whenever a tuple of a particular table is added/modified or when a foreign key or primary key has updated the integrity of referential constraint is violated. By default, SQL will restrict the update and avoid these violations, schema designer will specify what actions can be done and these are referential triggered actions. CASCADE, SET NULL, SET DEFAULT clauses are used whenever ON DELETE or ON UPDATE operations occur on tuples.

ex: A database stores "User Information" with "userEmail" of a user as a primary key and when a user updates their "userEmail(ON UPDATE) then the CASCADE clause is performed and all relations are updated with new "userEmail".

Chapter-6 Review

Yogesh Kalapala
yxk9640
1001879640

Q3
Sol

```
1.  CREATE TABLE BOOK
    (
    Book_id                 INT(5)          NOT NULL
    Title                   CHAR(10)        NOT NULL
    Publisher_name          CHAR(5)         NOT NULL
    - PRIMARY KEY(Book_id)
    - FOREIGN KEY (Publisher_name) REFERENCES PUBLISHER(Name)
    ON UPDATE CASCADE;
    )
2.  CREATE TABLE BOOK_AUTHORS
      (
    Book_id                 INT(5)          NOT NULL
    Author_name                     CHAR(10)        NOT NULL
    - PRIMARY KEY(Author_name, Book_id)
    - FOREIGN KEY(Book_id) REFERENCES BOOK(Book_id) ON DELETE
    CASCADE ON UPDATE CASCADE
    );
3.  CREATE TABLE PUBLISHER
      (
    Name                    CHAR(5)         NOT NULL
    Address                 CHAR(10)
    Phone                   INT(10)
    - PRIMARY KEY(Name)
    )
4.  CREATE TABLE BOOK_COPIES
      (
    Book_id                 INT(5)          NOT NULL
    Branch_id               VARCHAR(9)      NOT NULL
    No_of_copies            INT(2)
    - PRIMARY KEY(Book_id, Branch_id)
    - FOREIGN KEY(Book_id) REFERENCES BOOK(Book_id) ON DELETE
    CASCADE ON UPDATE CASCADE
    - FOREIGN KEY(Branch_id) REFERENCES
    LIBRARY_BRANCH(Branch_id) ON DELETE CASCADE ON UPDATE
    CASCADE
    );
```

Yogesh Kalapala
yxk9640
1001879640

5. CREATE TABLE BOOK_LOANS

```
(
Book_id                 INT(5)          NOT NULL
Branch_id               VARCHAR(9)      NOT NULL
Card_no                 INT(15)         NOT NULL
Date_out                DATE       NOT NULL
Due_date                DATE       NOT NULL
- PRIMARY KEY(Book_id, Branch_id,Card_no)
- FOREIGN KEY(Book_id) REFERENCES BOOK(Book_id) ON DELETE
CASCADE ON UPDATE CASCADE
- FOREIGN KEY(Branch_id) REFERENCES
LIBRARY_BRANCH(Branch_id) ON DELETE CASCADE ON UPDATE
CASCADE
- FOREIGN KEY(Card_no) REFERENCES BORROWER(Card_no) ON
DELETE CASCADE ON UPDATE CASCADE
)
```

6. CREATE TABLE LIBRARY_BRANCH

```
  (
Branch_id               VARCHAR(9)              NOT NULL
Branch_name             CHAR(10)        UNIQUE
Address                 CHAR(20)        NOT NULL
PRIMARY KEY(Branch_id)
)
```

7. CREATE TABLE BORROWER

```
  (
Card_no             INT(15)         NOT NULL
Name                CHAR(10)        NOT NULL
Address             CHAR(20)        NOT NULL
Phone               INT(10)         NOT NULL
PRIMARY KEY(Card_no)
)
```

Yogesh Kalapala
yxk9640
1001879640

Q4

Sol:

For Key constraint, When a tuple is added to the table the DBMS checks all the tuples of the tables and verifies if the key attribute of the tuple added is unique and only then the tuple is added. If not, the update clause is terminated.

For foreign Key constraint, If any tuple is added to the reference table the DBMS makes sure that the key(foreign) is also updated in all the referring tables with the new value.

Similarly, when a DELETE clause is implemented, the DBMS checks if, even after deleting the table's reference key, the tables referring to that tuple can be determined efficiently.

```
CREATE TABLE <Table Name>
CONSTRAINT PRIMARY KEY ()
CONSTRAINT FOREIGN KEY()  REFERENCES  PRIMARY
 KEY()
```

Q5

Sol:

The update command is used to change already existing attribute values to a new value. UPDATE is implemented with a combination of SET and WHERE clauses.

Ex: In a DB "PERSON" which stores user information ssn, name, city, age, and when a user changes the city we can use the UPDATE command to implement necessary changes

```
Update PERSON SET City = "Arlington" where ssn=
 712838429.
```

Chapter-6 Review

Yogesh Kalapala
yxk9640
1001879640

Q6
Sol:

PRIMARY KEY CLAUSE	UNIQUE CLAUSE
A table can have only 1 primary key	A table can have multiple unique keys
Primary clause impose entity integrity	Unique clause impose data to be unique
Does not allow null attribute values	Can have null attribute values

Q7
Sol

A. SELECT Name, Title FROM BORROWER, BOOK WHERE BOOK_LOANS.card_no=BOOK_LOANS.Book_id
B. SELECT Author_name, Title FROM BOOK_AUTHORS, FROM BOOK WHERE BOOK_AUTHORS.Author_name = BOOK_AUTHORS.Book_id
C. SELECT Name FROM PUBLISHER WHERE BOOK.Publisher=BOOK_AUTHORS.Author_name AND BOOK_AUTHORS.Author_name = "jk rowling”
D. SELECT Address FROM LIBRARY_BRANCH WHERE LIBRARY_BRANCH.Branch_id=BOOK_COPIES.Branch_id AND BOOK_COPIES.Branch_id= BOOK.Book_id AND BOOK.Title = "Don Quixote”.

Q8
Sol

A. SELECT Name FROM STUDENT WHERE STUDENT.Major = “CS".
B. SELECT Course_name FROM COURSER,SECTION WHERE SECTION.Course_number=COURSE.Course_number AND SECTION.Instructor= "King" AND (SECTION.Year=2007 OR SECTION.Year=2008).

Yogesh Kalapala
yxk9640
1001879640

C. SELECT course_number,semester, year,count(STUDENT.Student_number) FROM SECTION,GRADE_REPORT WHERE SECTION.Instructor = "King" AND SECTION.Identifier = GRADE_POINT.Identifier
D. SELECT name,Grade,Course_name,Course_number,Credit_hour, Semester,Year FROM STUDENT,GRADE_REPORT, COURSE, SECTION WHERE STUDENT.class = 4 AND Major = CS AND STUDENT.Student_number = GRADE_REPORT.Student_number AND COURSE.Course_number = SECTION.Course_number AND SECTION.Section_Identifier = GRADE_REPORT.Section_Identifier

Q9
Sol

A. INSERT Name,Student_number,Class,Major INTO STUDENT values('Johnson',25,1,Math)
B. UPDATE STUDENT SET class=2 WHERE Name='Smith'
C. INSERT Course_name,Course_number,Credit_hours,Department INTO COURSE values('Knowledge Engineering',cs4390,3,'cs')
D. DELETE * FROM STUDENT WHERE Name=Smith AND Student_number=17