CSCI 3901

**Software Development Concepts**



Faculty of Computer Science

**Lab 9: “Database Design”**

Kishan Kahodariya B00864907

Dhruv Patel B00868931

**Part – 01 Key Constraints**

**Make following changes to the databases:**

1. **Add an entry to course with web id of NULL**

* insert into course value (4,” csci4521”, null);

1. **Add an entry to course with web id of 2**
   * insertinto course value (5,” csci4420”, 2);
2. **Add an entry to course with web id of 4**
   * insertinto course value (6,” csci4720”, 4);
3. **Add an entry to web with web id of 5**
   * insertinto web value (5, ” [www.slack.com](http://www.slack.com)”);

**Explain how you would identify all the entries with bad web\_id keys.**

* Select c.web\_id from course c lest join web w on c.web\_id=w.web\_id

where w.web\_id is null AND c.web\_id is not null;

**Determine whether the foreign key constraint can be successfully added with the above command in each of the following cases:**

1. **course contains a web id that is a bad foreign key**
   * It fails because foreign key constraint fails
2. **course contains a web id that is NULL**
   * It will be successfully added
3. **course contains only web ids that are in web**
   * It will be successfully added

**Add web id as a foreign key in course, making any changes to either table necessary to do so.**

* You can either update or either delete the bad web\_id foreign key and then add it to course table.
  1. update web set web\_id=4 where web\_id=5;
  2. alter table course add foreign key (web\_id) references web (web\_id)

**Explain what it means if you allow a foreign key column to be NULL**

* Course table will allow adding a column as foreign\_key having NULL from other table if that column is allowed to have NULL value.

**Explain what it means if you do not allow a foreign key column to be NULL**

* For every course there exists web\_id in web table which means no new course can be added unless there is corresponding web\_id.

**Make the following changes to the database and report on their success or failure:**

1. **Delete course key 2 in course**
   * delete from course where course\_key=2
   * This query will succeed
2. **Delete web id 3 in web**
   * delete from web where web\_id=3
   * This query will not succeed
3. **Delete web (i.e. using drop table web;)**
   * drop table web;
   * This query will not succeed
4. **Delete course (i.e. using drop table course**;)
   * drop table course;
   * This query will succeed

**Explain why the previous commands succeeded or failed.**

1. This query succeeds as course table doesn’t contain any foreign key references.
2. This query fails as this will have bad foreign key i.e. web\_id=3 in course table.
3. This query fails as it will also delete foreign key references from course table.
4. We can delete course table as it doesn’t have any foreign key references from other table.

**Part – 02 Database Design from ERD**

**SQL Query**

create table ds\_EVENT(

Event\_ID int primary key,

Event\_Date date,

Event\_Location varchar(20),

Event\_Time time

);

create table ds\_MENU(

Menu\_ID int primary key,

Menu\_description varchar(20),

Menu\_type varchar(20)

);

create table ds\_DISH(

Dish\_ID int primary key,

Dish\_Name varchar(20),

Prep\_Time time

);

create table ds\_DISH\_Ingredient(

Ingredient varchar(20),

Dish\_id int NOT NULL REFERENCES ds\_DISH(Dish\_ID) ON DELETE cascade

);

describe ds\_DISH\_Ingredient;

create table ds\_WORK\_SCHEDULE(

Event\_ID int,

Emp\_ID int,

Start\_Time time,

End\_Time time,

Position time,

foreign key(Event\_ID) references ds\_EVENT(Event\_ID),

foreign key(Emp\_ID) references ds\_STAFF(Emp\_ID)

);

create table ds\_STAFF(

Emp\_ID int primary key,

Supervisor\_ID int,

Staff\_Name varchar(20),

Salary int,

foreign key(Supervisor\_ID) references ds\_STAFF(Emp\_ID)

);

create table ds\_STAFF\_Skill(

Skill varchar(20),

Emp\_ID int NOT NULL REFERENCES

ds\_STAFF(Emp\_ID) ON DELETE cascade

);

alter table ds\_WORK\_SCHEDULE add foreign key (Event\_ID) references ds\_EVENT(Event\_ID);

alter table ds\_WORK\_SCHEDULE modify Event\_ID int not null;

alter table ds\_WORK\_SCHEDULE add foreign key (Emp\_ID) references ds\_STAFF(Emp\_ID);

alter table ds\_WORK\_SCHEDULE modify Emp\_ID int not null;

alter table ds\_EVENT add Menu\_ID int;

alter table ds\_EVENT add foreign key (Menu\_ID) references ds\_MENU(Menu\_ID);

alter table ds\_EVENT modify Menu\_ID int not null;

create table ds\_Contains(

Menu\_ID int,

Dish\_ID int

);

alter table ds\_Contains add foreign key (Menu\_ID) references ds\_MENU(Menu\_ID);

alter table ds\_Contains modify Menu\_ID int not null;

alter table ds\_Contains add foreign key (Dish\_ID) references ds\_DISH(Dish\_ID);

alter table ds\_Contains modify Dish\_ID int not null;

**Questions**

**How can foreign key constraints help to maintain the integrity of data in your database?**

* Foreign key constraints allow us to develop relationships among tables such as One to One, One to Many, Many to One, Many to Many. This constraint helps us to integrate data as per our requirement

​

**Is there only one valid design for a database with a given ER diagram?**

* No there can be multiple design according to what user interprets from ERD. For example, in our ERD we know that Dish contains multi valued attribute Ingredient but is this attribute one to many or many to many is not specified, these can create ambiguity as many ingredients may also present in dish