

Lab 9: Database Design

Team Members:

1. Dhrumil Amish Shah
[REDACTED]
2. Dhrumil Rakesh Shah
[REDACTED]

Date:

2021-03-24

—

Subject:

Software Development Concepts

—

Professor:

Matthew Amy

Part 1: Key Constraints

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

- a. Add an entry to course with web id of NULL.

Success

	course_key	name	web_id
▶	1	csci3901	1
	2	csci5100	1
	3	math1000	3
	4	csci4000	NULL
•	NULL	NULL	NULL

- b. Add an entry to course with web id of 2.

Success

	course_key	name	web_id
▶	1	csci3901	1
	2	csci5100	1
	3	math1000	3
	4	csci4000	NULL
	5	csci5000	2
•	NULL	NULL	NULL

- c. Add an entry to course with web id of 4.

Success

	course_key	name	web_id
▶	1	csci3901	1
	2	csci5100	1
	3	math1000	3
	4	csci4000	NULL
	5	csci5000	2
	6	csci9000	4
•	NULL	NULL	NULL

- d. Add an entry to web with web id of 5.

Success

	web_id	URL
▶	1	cs.dal.ca
	2	google.com
	3	dal.ca
	5	goolge.com
•	NULL	NULL

- Explain how you could identify all the entries in course with bad web id keys.

Web id is primary key in the web table, and it acts as a foreign key in the course table, so primary key should be unique. By running the below queries, I was able to identify the entries in course table with bad web_id keys.

a. Query 1:

`select web_id from course where web_id not in (select web_id from web);`

Result Grid	
	web_id
▶	4

b. Query 2:

```
select
  course.web_id
from
  course left join web on course.web_id = web.web_id
where
  web.web_id is null and
  course.web_id is not null;
```

Result Grid	
	web_id
▶	4

- A foreign key constraint can be added to the course table with the command:

alter table course add foreign key (web_id) references web (web_id);

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

- a. course contains a web id that is a bad foreign key.

Failed

The course table cannot be altered with the above-mentioned command with web id as a bad foreign key is present in course table.

- b. course contains a web id that is NULL.

Success

The course table can be altered with web id being NULL.

- c. course contains only web ids that are in web.

Success

The alter command succeeds when course contains only web ids that are in the web table.

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

Run the below query, to make the necessary changes.

update web set web_id = 4 where web_id = 5;

After running the above query, I am able to add web_id as a foreign key in course table by running the below query.

alter table course add foreign key (web_id) references web (web_id);

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

Allowing null values in foreign key column can degrade its referential integrity, since any key with a null value is never checked against the referenced table. To avoid this, we can use a NOT NULL constraint on foreign keys when creating the tables. But if we allow a foreign key to be NULL then its okay as it a key whose value should exist in the referenced table.

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

If we do not allow a foreign key to be NULL, then it means that we are properly following the referential integrity and using a NOT NULL constraint on foreign keys while creating tables. It does not make much of a difference if we do not allow NULL values as foreign key as long as the foreign key is present in the referenced table.

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

- a. Delete course key 2 in course.

Success

delete from course where course_key = 2;

	course_key	name	web_id
▶	1	csci3901	1
	3	math1000	3
	4	csci4000	NULL
	5	csci5000	2
	6	csci9000	4
★	NULL	NULL	NULL

- b. Delete web id 3 in web.

Failed

We cannot delete web_id = 3 as web_id is a foreign key in course table to if we delete web_id = 3 row from web table then it will violate the foreign key constraint.

delete from web where web_id = 3;

- c. Delete web (i.e., using drop table web;)

Failed

We cannot drop the web table as web_id is a foreign key in course table, so deleting web table will lead to violation of foreign key constraint.

drop table web;

- d. Delete course (i.e., using drop table course;)

Success

We can drop the course table as it has no foreign key constraint with the web table, so there arises no conflict.

drop table course;

8. Explain why the previous commands succeeded or failed.

The previous commands succeeded and failed because of the foreign key constraint between the tables. There exists a constraint between the course table and web table. The course table has web_id as a foreign key. So, to maintain the foreign key consistency, the values of web_id should match between both the tables. We cannot delete web table nor can we delete any web_id value from the web table. But on the other side, we can delete the course table without affecting the web table.

Part 2: Database Design

```
create table ds_MENU (  
Menu_ID int primary key not null auto_increment,  
Menu_Description varchar(50) not null,  
Menu_Type varchar(25) not null);
```

```
create table ds_DISH (  
Dish_ID int primary key not null auto_increment,  
Dish_Name varchar(25) not null,  
Prep_Time time not null);
```

```
create table ds_MENU_DISH (  
Menu_ID int not null,  
Dish_ID int not null,  
primary key (Menu_ID,Dish_ID),  
foreign key (Menu_ID) references ds_MENU (Menu_ID),  
foreign key (Dish_ID) references ds_DISH (Dish_ID));
```

```
create table ds_INGREDIENT (  
Ingredient_ID int primary key not null auto_increment,  
Ingredient_Name varchar(25) not null);
```

```
create table ds_DISH_INGREDIENT (  
Dish_ID int not null,  
Ingredient_ID int not null,  
primary key (Dish_ID,Ingredient_ID),  
foreign key (Dish_ID) references ds_DISH (Dish_ID),  
foreign key (Ingredient_ID) references ds_INGREDIENT (Ingredient_ID));
```

```
create table ds_EVENT (  
Event_ID int primary key not null auto_increment,  
Event_Date date not null,  
Event_Location varchar(25) not null,  
Event_Time time not null,  
Menu_ID int not null,  
foreign key (Menu_ID) references ds_MENU (Menu_ID));
```

```
create table ds_STAFF (  
Emp_ID int primary key not null auto_increment,  
Staff_Name varchar(25) not null,  
Salary int not null,  
Supervisor_ID int,  
foreign key (Supervisor_ID) references ds_STAFF (Emp_ID));
```

```
create table ds_WORK_SCHEDULE (
Start_Time time not null,
End_Time time not null,
Position varchar(25) not null,
Event_ID int not null,
Emp_ID int not null,
foreign key (Event_ID) references ds_EVENT (Event_ID),
foreign key (Emp_ID) references ds_STAFF (Emp_ID),
primary key(Position,Event_ID, Emp_ID));
```

```
create table ds_Skill (
Skill_ID int primary key not null auto_increment,
Skill_Name varchar(25) not null);
```

```
create table ds_STAFF_SKILL (
Emp_ID int not null,
Skill_ID int not null,
primary key (Emp_ID,Skill_ID),
foreign key (Emp_ID) references ds_STAFF (Emp_ID),
foreign key (Skill_ID) references ds_Skill (Skill_ID));
```

Questions

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

Foreign keys help maintain the consistency of our data, and it provides a certain measure of convenience. Without foreign keys, we are responsible for keeping track of inter-table dependencies and maintaining their consistency from within our applications. It amounts to little more than adding a few extra DELETE statements to make sure that when we delete a record from one table, we also delete the corresponding records in any related tables. But if our tables have particularly complex relationships, we may not want to be responsible for implementing these dependencies in your applications.

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

No, there can be multiple valid database designs for the given ER diagram.

For example

In our case, we have considered the below query to create a work schedule table.

Here we are considering (Position, Event_ID, Emp_ID) as Primary Key.

```
create table ds_WORK_SCHEDULE (
Start_Time time not null,
End_Time time not null,
Position varchar(30) not null,
Event_ID int not null,
```

```
Emp_ID int not null,  
foreign key (Event_ID) references ds_EVENT (Event_ID),  
foreign key (Emp_ID) references ds_STAFF (Emp_ID),  
primary key(Position,Event_ID, Emp_ID)  
);
```

But another possible solution could be as per the below query to create a work schedule table.
Here we are considering (Start_Time, End_Time, Event_ID, Emp_ID) as Primary Key

```
create table ds_WORK_SCHEDULE (  
Start_Time time not null,  
End_Time time not null,  
Position varchar(30) not null,  
Event_ID int not null,  
Emp_ID int not null,  
foreign key (Event_ID) references ds_EVENT (Event_ID),  
foreign key (Emp_ID) references ds_STAFF (Emp_ID),  
primary key(Start_Time, End_Time, Event_ID, Emp_ID)  
);
```