

ACIT 1630 - Relational Database Design and SQL

Intro to Primary and Foreign Keys in MySQL Server

ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys

Introduction:

The goal of this document is to serve as a walkthrough for the creating and managing primary and foreign keys in MySQL Server (MySQL).

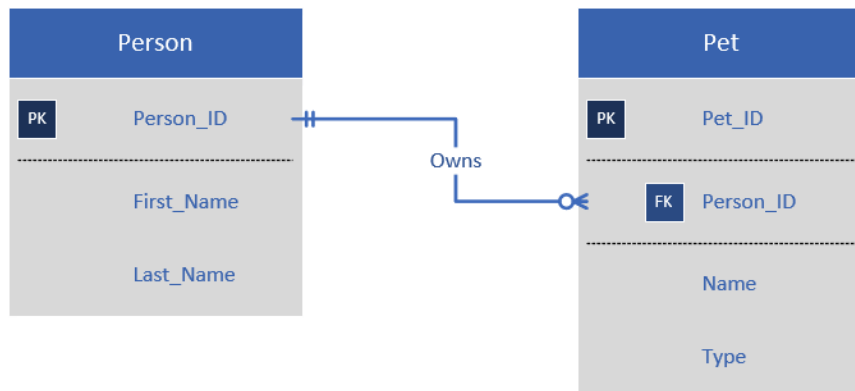
Primary and Foreign Keys are what link tables (and the entities they represent) together. Without Primary and Foreign Keys, our relational database wouldn't be *relational*.

For a quick definition, a Foreign Key is just a reference to a Primary Key in another table. Remember that a Primary Key needs to have a few qualities to make it a good Primary Key:

- Must be Unique
- Cannot be NULL (Allows Nulls = No)
- Should not change
- Often autogenerated (is identity = Yes)

As you'll discover with this Lab, the order in which you do things will matter. Since a Foreign Key refers to the Primary Key, the Primary Key should be created *first*.

In this Lab we are going to implement our Person, Pet One-to-Many Relationship with Primary and Foreign Keys.



Step 1:

Open MYSQL Workbench

ACIT 1630 Relational Database Design and SQL
Intro to Primary and Foreign Keys

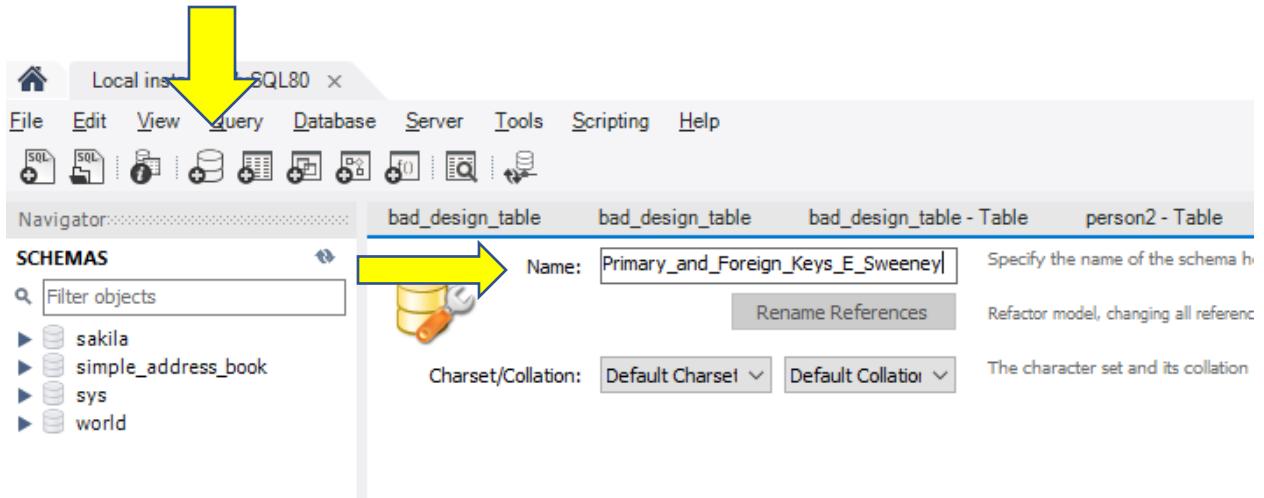
Step 2:

Create a new schema in the following format:

Primary_and_Foreign_Keys_ + Your First Initial + _ + Your Last Name

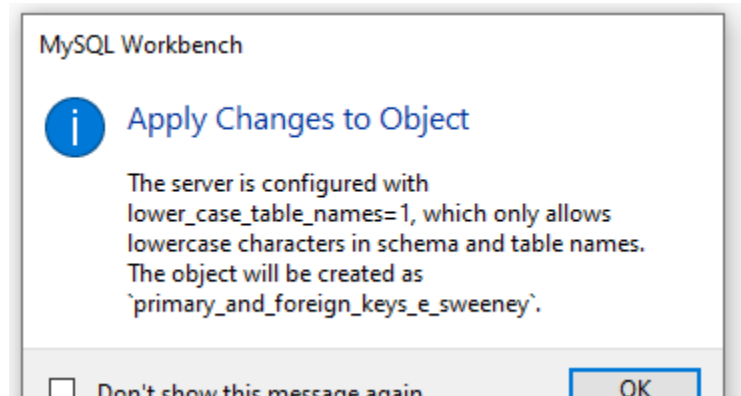
As an example, for me I would create the database as:

Primary_and_Foreign_Keys_E_Sweeney



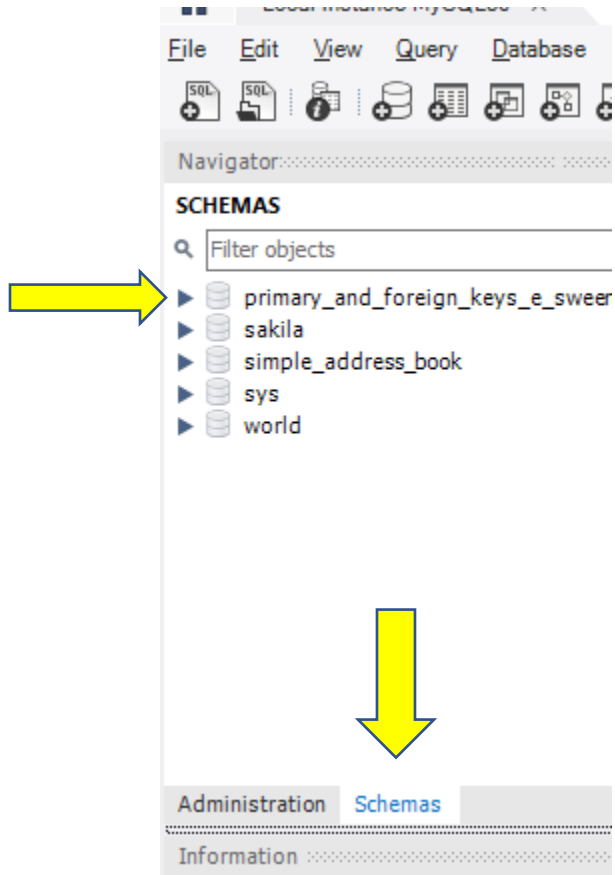
Click Apply

You will see a notice that MySQL doesn't like capital letters in its schema and table names, so it will lowercase all.



ACIT 1630 Relational Database Design and SQL
Intro to Primary and Foreign Keys

After clicking OK you will see the schema listed in the left side navigator, if you have select the schemas tab on bottom of the navigator pane



ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys

*Note: pages 5 to 11 are for illustrative purposes only, you should not do these steps. Continue lab work on page 12

Testing the importance of a Primary Key

Before we get into creating our Person and Pet tables, let's create a table that doesn't have a Primary Key. A table without a Primary Key can cause troubles as we'll soon see.

Create the following table in your new database with the following columns:

bad_design_1:

Table Name: bad_design_table Schema: primary_and_secondary

Charset/Collation: De De Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN
first_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
last_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
grade	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Column Name: type: INT

Charset/Collation: default

Comments: age: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert Context Help

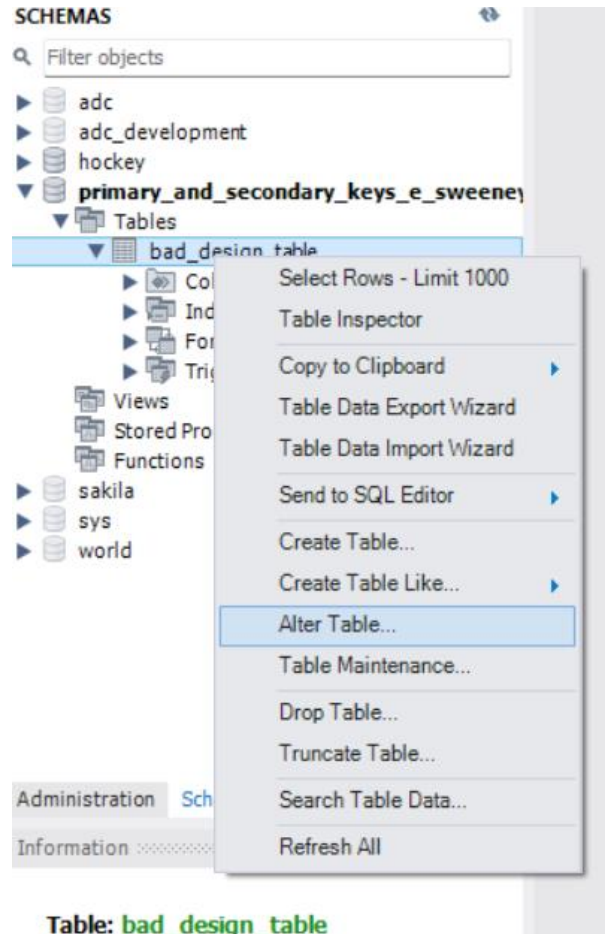
Let's say there are 3 people in our fictitious class:

First_name	Last_name
Zhang	Cheng
Mary	Silva
Zhang	Cheng

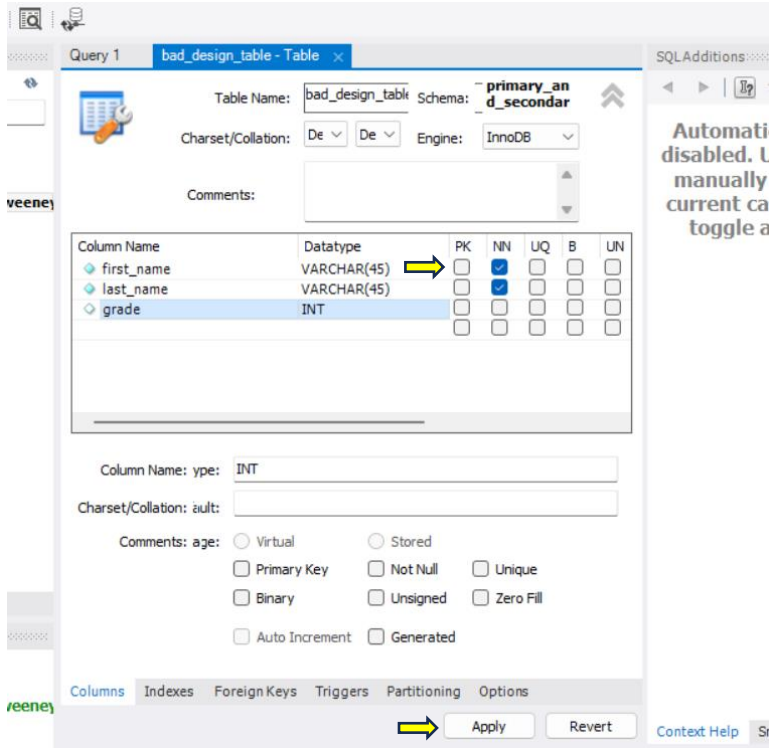
ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys

Go ahead and enter these 3 people in our table – but first, because it is such a bad design, MySQL Workbench won't even allow us to use the table data editor, so we have to make one of the fields the primary key, **temporarily**, just to add the data. We do this with the Alter table functionality and clicking the PK box for first name:



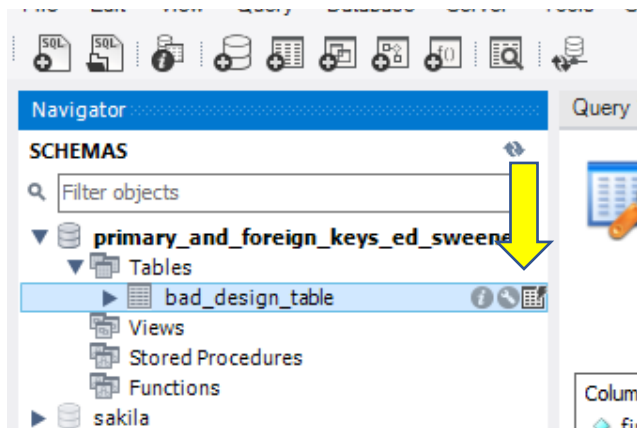
Intro to Primary and Foreign Keys



After Applying this table alter statement (executing the sql), we can now enter this data:

First_name	Last_name
Zhang	Cheng
Mary	Silva
Zhang	Cheng

We enter data with the table data editor:



ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys

Tools Scripting Help

Query 1 contacts - Table primary_and_foreign_keys_ed... bad_design_table - Table x

Table Name: bad_design_table Schema: primary_and_fore

Charset/Collation: Default Charset Default Collation Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
first_name	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
last_name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
grade	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: Data Type:

Charset/Collation: Default Charset Default Collation Expression:

Comments: Storage:

We will leave the “g” letter off of the third record’s first name for now, to make it unique, and apply the data changes

Result Grid Filter Rows: Edit:

	first_name	last_name	grade
	Zhang	Cheng	NULL
	Mary	Silva	NULL
	Zhan	Cheng	NULL
*	NULL	NULL	NULL

Result Grid

Form Editor

Field Types

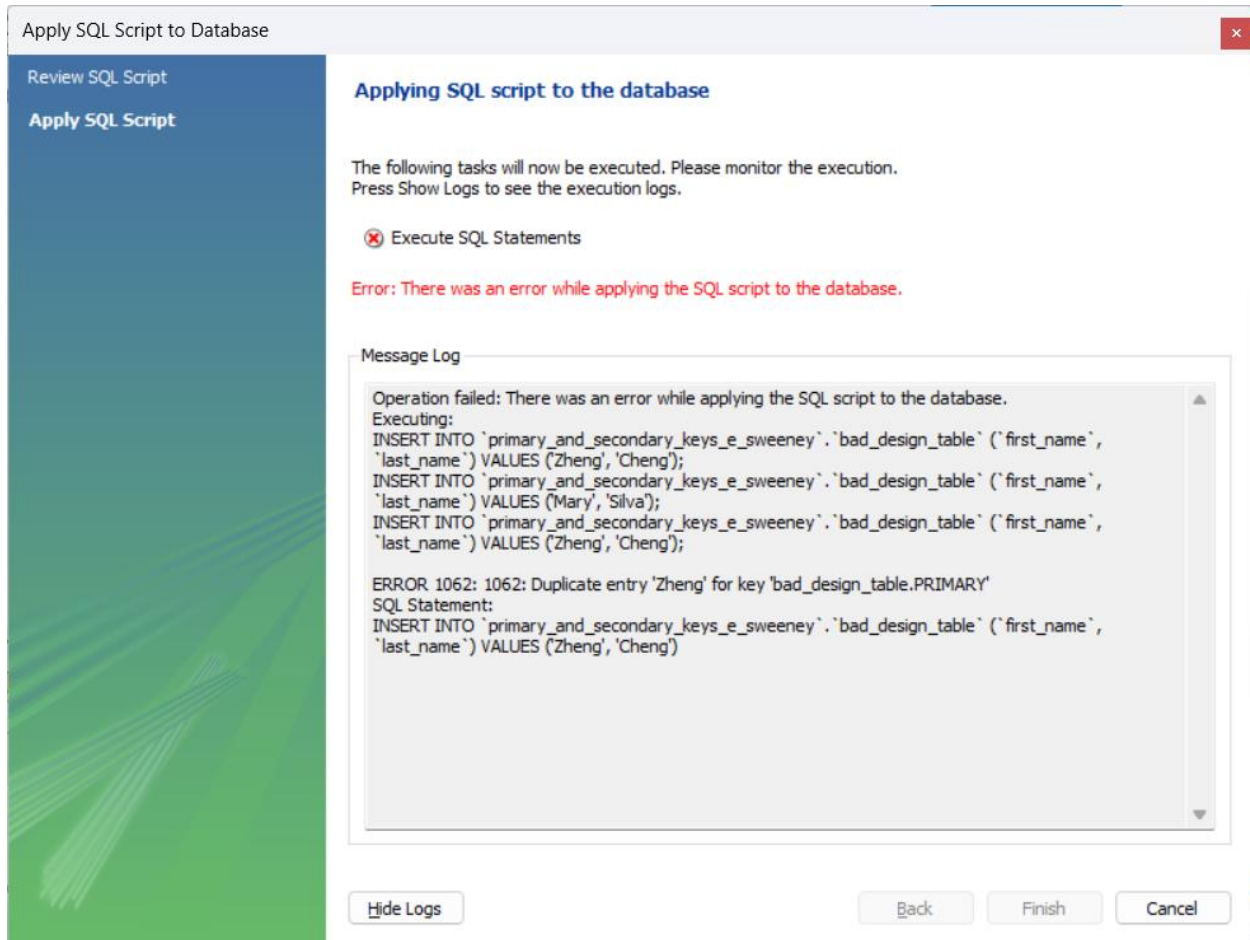
design_table1 x Apply Revert Conte

ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys

[optional]

Just to prove a point we can go back to the data entry editor, change the third record back to “Zhang” and try to save it even though we know it is not unique



When MySQL was executing the statement to enter the data, it threw up because we tried to enter the same value for two different record's primary keys. This is to validate that all of our primary keys are unique.

Now we go back to the Alter Table window, deselect the first_name primary key designation and apply.

Then go back to the data entry tab and change the first name back to “Zhang” and apply – it should work this time because the field is no longer a primary key, nor is it designated as requiring a unique value (for any other reason).

But ALAS, MySQL now throws up the exception:

ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys


Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Applying SQL script to the database

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

 Execute SQL Statements

Error: There was an error while applying the SQL script to the database.

Message Log

```
Operation failed: There was an error while applying the SQL script to the database.  
Executing:  
UPDATE `primary_and_foreign_keys_ed_sweeney`.`bad_design_table` SET `first_name` = 'Zhang'  
WHERE (`first_name` = 'Zhan');  
  
ERROR 1175: 1175: You are using safe update mode and you tried to update a table without a WHERE  
that uses a KEY column.  
SQL Statement:  
UPDATE `primary_and_foreign_keys_ed_sweeney`.`bad_design_table` SET `first_name` = 'Zhang'  
WHERE (`first_name` = 'Zhan')
```

Hide Logs

Back

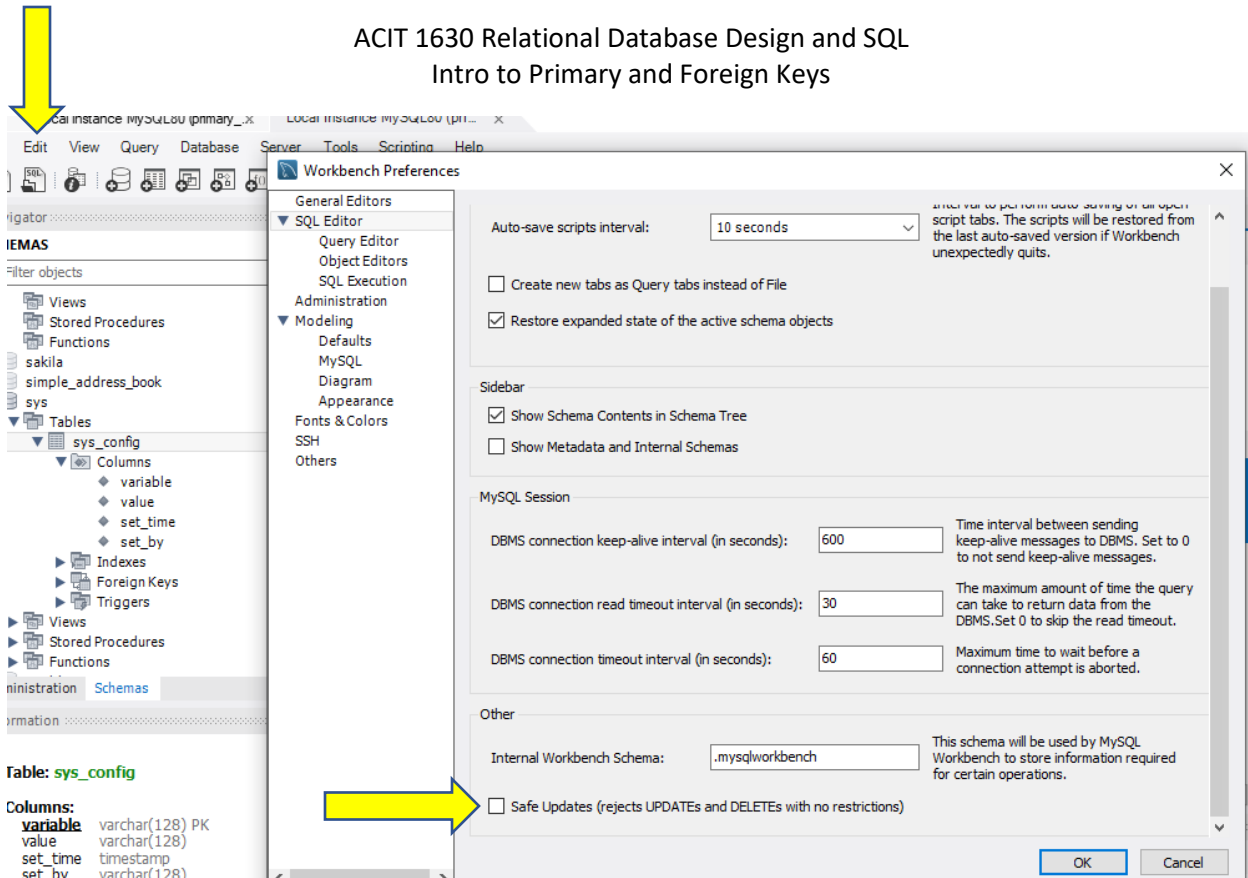
Finish

Cancel

It really doesn't like poor design! If we are really determined to make poor choices we can change the change the safe update option here:

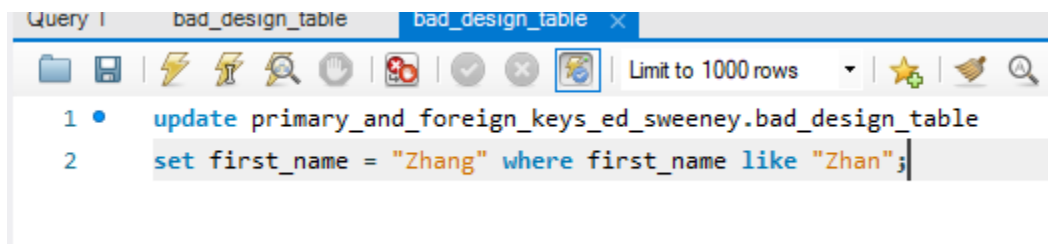
ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys



Scroll down if you can't see the safe updates option, then apply. Then repeat the data value change (above) for "Zhang" and apply.

Then you would have to restart your Workbench because the options are set at application start time.



Now let's go and try to update their grades.

Let's give Zhang Cheng a grade of 75%.

Uh, oh! Right away, MySQL notices something wrong and gives us an error message.

The error message says some confusing things, but, is basically trying to say "Which Zhang Cheng should I modify? – I found 2!!". Without a primary key to enforce every row is unique, MS SQL is afraid to modify a single row, because it thinks it might affect more than one entry. Yikes!

This is why Primary Keys are so important!

ACIT 1630 Relational Database Design and SQL
Intro to Primary and Foreign Keys

***Continue your step by step lab work here

Step 2:

Create the following tables in your new database with the following columns:

Person:

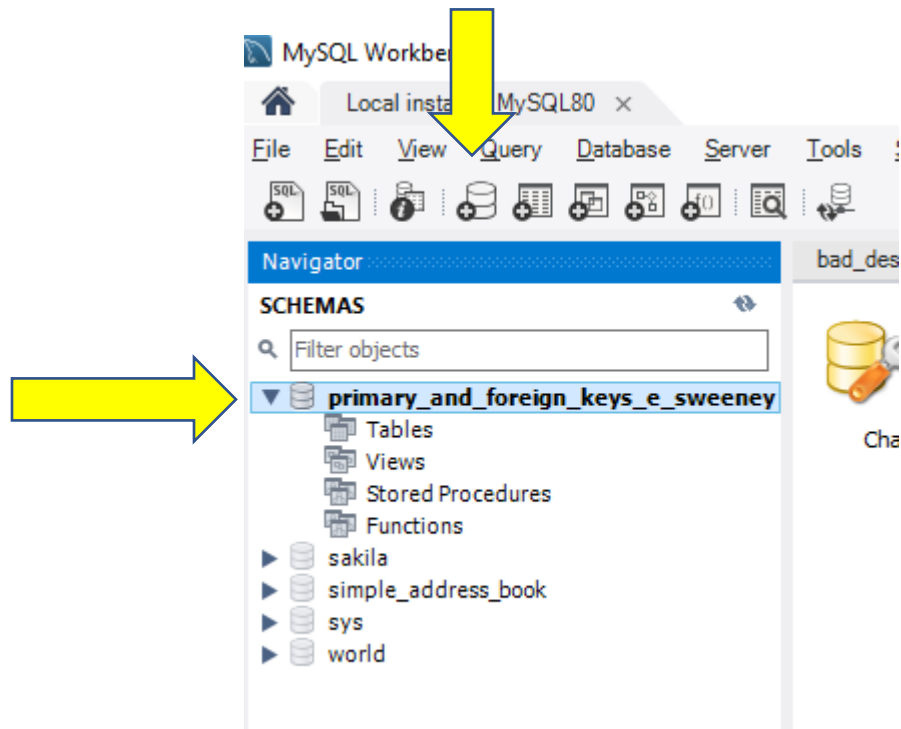
Column Name	Data Type	Allow Nulls	Is Identity	Key
person_id	int	No	Yes	Primary
first_name	nvarchar(50)	No		
last_name	nvarchar(50)	No		

Pet:

Column Name	Data Type	Allow Nulls	Is Identity	Key
pet_id	int	No	Yes	Primary
person_id	Int	No		Foreign Key to Person
name	nvarchar(50)	No		
type	nvarchar(50)	No		

We confirm that the correct schema is being used by double clicking on the name in the navigator window, which should make it appear in **BOLD** text

Then we click on the menu icon that appears to be a cylinder



ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys

Next we use the alter table tool to add the required fields and options

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
person_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
first_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
last_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Remember to add our Primary Key, NN (not null) and AI checked options as above.

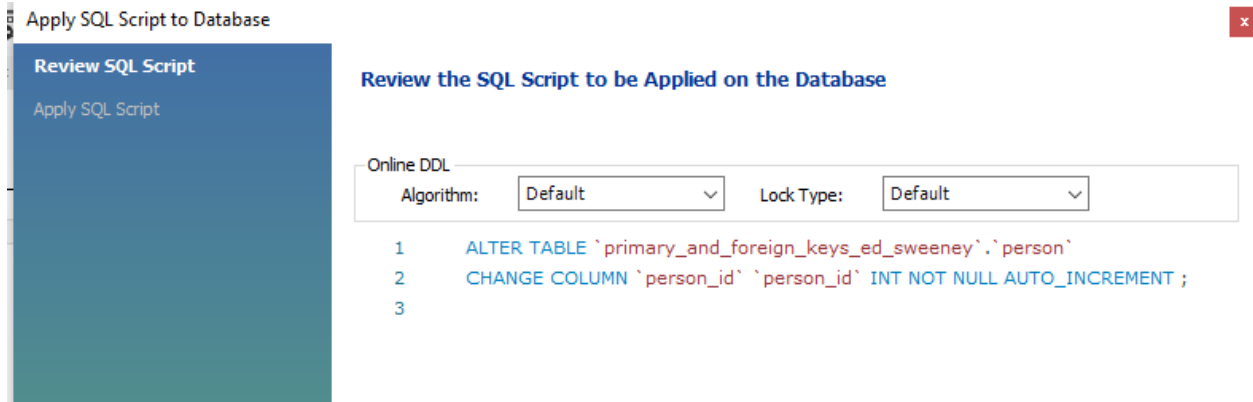
AI is not artificial intelligence, in this case. To make our lives a bit easier, we are going to tell MySQL to manage the creation of the primary key value for us, so we don't have to keep track of which number we should use next. MySQL has a convenient option called Auto Increment, which, if selected, will allow the system to determine the next key value to use. We do this by altering the Person table, and checking the AI box for the primary key field:


Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
person_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
first_name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
last_name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL




Click apply.

ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys



Now we should see a key icon  beside our person_id column definition.

Column Name	Datatype	PK	NN	I
 person_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	[
 first_name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	[
 last_name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	[

Create the Pet table in much the same way as our Person table.

Don't forget to make the pet_id as a Primary Key.

If you do it correctly and click apply, the confirmation screen should look like this:

ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys

Apply SQL Script to Database



Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

Online DDL

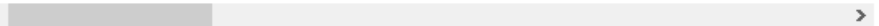
Algorithm:

Default

Lock Type:

Default

```
1 CREATE TABLE `primary_and_foreign_keys_ed_sweeney`.`pet` (  
2   `pet_id` INT NOT NULL,  
3   `person_id` INT NULL,  
4   `name` VARCHAR(45) NULL,  
5   `type` VARCHAR(45) NULL,  
6   PRIMARY KEY (`pet_id`));  
7
```



Back

Apply

Cancel

ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys

Now we need to create a Foreign Key on the person_id.

The screenshot shows the 'Foreign Key' configuration window in MySQL Workbench. The 'Table Name' is 'pet' and the 'Schema' is 'primary_and_foreign_keys_ed_sweeney'. The 'Charset/Collation' is set to 'utf8mb4' and 'utf8mb4_0900_ai_ci', and the 'Engine' is 'InnoDB'. The 'Comments' field is empty.

Foreign Key Name	Referenced Table
person_id	'primary_and_foreign_keys_ed_sweeney'.

Column	Referenced Column
<input type="checkbox"/> pet_id	
<input checked="" type="checkbox"/> person_id	person_id
<input type="checkbox"/> name	
<input type="checkbox"/> type	

Foreign Key Options:

- On Update: NO ACTION
- On Delete: NO ACTION
- ☐ Skip in SQL g

Foreign Key Comment:

At the bottom, there are tabs for 'Columns', 'Indexes', 'Foreign Keys' (selected), 'Triggers', 'Partitioning', and 'Options'. An 'Apply' button is located at the bottom right.

The Pet table will be your base table, select person_id as the column.

Select Person table as the Primary Key table and the person_id column.

Click Apply and Apply again.

The screenshot shows the 'Review SQL Script to be Applied on the Database' dialog box. It displays the following SQL script:

```
1 ALTER TABLE `primary_and_foreign_keys_ed_sweeney`.`pet`
2 ADD INDEX `person_id_idx` (`person_id` ASC) VISIBLE;
3 ;
4 ALTER TABLE `primary_and_foreign_keys_ed_sweeney`.`pet`
5 ADD CONSTRAINT `person_id`
6 FOREIGN KEY (`person_id`)
7 REFERENCES `primary_and_foreign_keys_ed_sweeney`.`person` (`person_id`)
8 ON DELETE NO ACTION
9 ON UPDATE NO ACTION;
10
```

At the bottom, there are 'Back', 'Apply', and 'Cancel' buttons.

ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys

We have just created our first Foreign Key!! What this will do is ensure that every pet has an owner (and that owner already exists in our database)! This is a worthy goal, right?!

Let's test it out.

Let's try to add Rango the Dog to our Pet table. Since we set our person_id to 7, which we know does not exist in our person table. It's a mistake we would rather avoid, and MySQL won't let us because we have added the foreign key constraint.

The screenshot shows a database application interface. At the top, there's a 'Result Grid' tab with a table containing columns: pet_id, person_id, name, and type. The first row has values 1, 7, Rango, and Dog. Below it, there's a row with NULL values. Below the table, there's a menu bar with 'base', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu bar, there's a tab 'Apply SQL Script to Database'. On the left, there's a sidebar with 'Review SQL Script' and 'Apply SQL Script'. The main area shows the title 'Review the SQL Script to be Applied on the Database' and a SQL script:

```
1 UPDATE `primary_and_foreign_keys_ed_sweeney`.`pet` SET `person_id` = '7' WH
2
```

The screenshot shows the same database application interface as before, but now the 'Apply SQL Script to Database' tab is active. The sidebar shows 'Review SQL Script' and 'Apply SQL Script'. The main area shows the title 'Applying SQL script to the database'. Below the title, there's a message: 'The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.' Below this, there's a button 'Execute SQL Statements' with a red 'X' icon. Below the button, there's a red error message: 'Error: There was an error while applying the SQL script to the database.' Below the error message, there's a 'Message Log' section. The log shows the following text:

```
Operation failed: There was an error while applying the SQL script to the database.
Executing:
UPDATE `primary_and_foreign_keys_ed_sweeney`.`pet` SET `person_id` = '7' WHERE (`pet_id` = '1');

ERROR 1452: 1452: Cannot add or update a child row: a foreign key constraint fails
(`primary_and_foreign_keys_ed_sweeney`.`pet`, CONSTRAINT `person_id` FOREIGN KEY
(`person_id`) REFERENCES `person` (`person_id`))
SQL Statement:
UPDATE `primary_and_foreign_keys_ed_sweeney`.`pet` SET `person_id` = '7' WHERE (`pet_id` = '1')
```

ACIT 1630 Relational Database Design and SQL

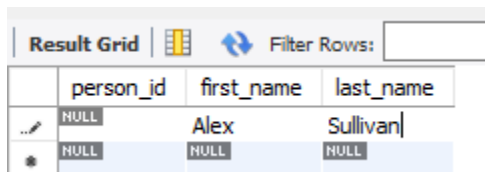
Intro to Primary and Foreign Keys

This doesn't work since we haven't added any people to the Person table with the `person_id` of 7.

Ok, so we've learned that we *have* to add people to the Person table *before* the pets in the Pet table. We'll also need to record their `person_id` that gets automatically generated for us so that we can refer to it in the Pet table.

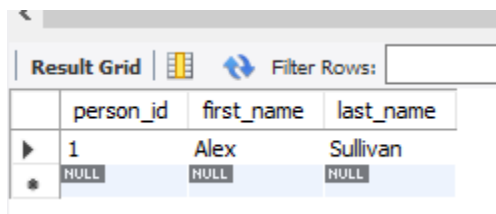
Let's try this again.

Add Alex Sullivan as a person in the Person table using the table data editor, we can leave the `person_id` field null and allow MySQL to set the key value because earlier we added the auto increment option.



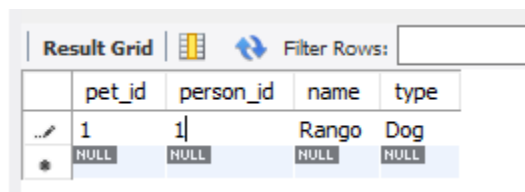
	person_id	first_name	last_name
...	NULL	Alex	Sullivan
*	NULL	NULL	NULL

Once we run that change we see the index value created by MySQL



	person_id	first_name	last_name
▶	1	Alex	Sullivan
*	NULL	NULL	NULL

Now we assign Rango to Alex as his human:



	pet_id	person_id	name	type
...	1	1	Rango	Dog
*	NULL	NULL	NULL	NULL

And Apply and...

Yay! Rango now has an owner!

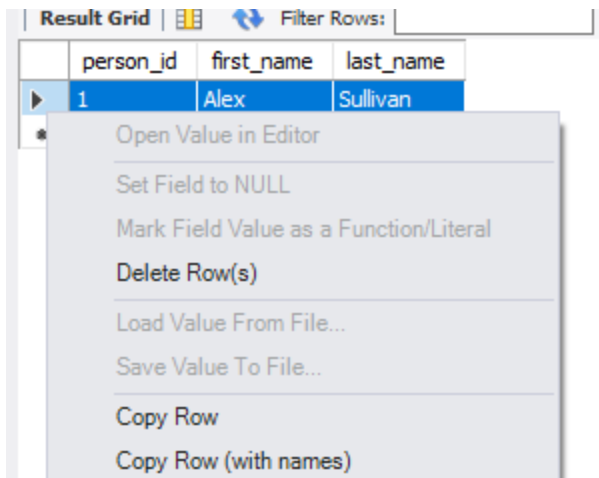
One more thing that Foreign Keys do for us is make sure that we don't accidentally delete Rango's owner.

Can you delete Alex without first deleting Rango? You shouldn't be able to. Otherwise we would again have a situation where Rango was ownerless.

Edit the Person table, click on the Alex row to select it, then right click on it and click Delete.

ACIT 1630 Relational Database Design and SQL

Intro to Primary and Foreign Keys

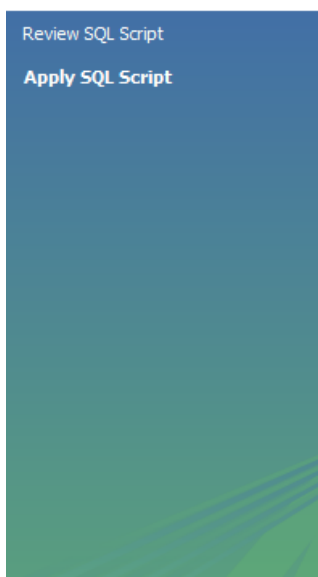


Click Apply

Say "Yes" to the "Are you sure you want to delete the row" message.


Again, MySQL will prevent us from leaving Rango without an owner – how humane of it!

Apply SQL Script to Database



Applying SQL script to the database

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

 Execute SQL Statements

Error: There was an error while applying the SQL script to the database.

Message Log

```
Operation failed: There was an error while applying the SQL script to the database.  
Executing:  
DELETE FROM `primary_and_foreign_keys_ed_sweeney`.`person` WHERE (`person_id` = '1');  
  
ERROR 1451: 1451: Cannot delete or update a parent row: a foreign key constraint fails  
(`primary_and_foreign_keys_ed_sweeney`.`pet`, CONSTRAINT `person` FOREIGN KEY (`person_id`)  
REFERENCES `person` (`person_id`))  
SQL Statement:  
DELETE FROM `primary_and_foreign_keys_ed_sweeney`.`person` WHERE (`person_id` = '1')
```

ACIT 1630 Relational Database Design and SQL
Intro to Primary and Foreign Keys

Step 3:

Go ahead and add a few more pets and owners to the database.

With Alex and Rango still in the database, add yourself to the Person table.

If you have any pets, add them to the Pet table.

Add a few other people and pets from the following table:

Katie Sylvia	Max (dog), Duke (dog)
Penny Superbark	Bolt (dog), Mittens (cat), Rhino (hamster)
Fix-it Felix	
Gru Despicable	Kyle (dog?)
<Your Name>	<Your Pets, if any>

Submit a screenshot of your **Person** table configuration (from the alter table editor).

Filename: **04_Person_Table.jpg**

Submit a screenshot of your **Pet** table configuration.

Filename: **05_Pet_Table.jpg**

Submit a screenshot of your **Pet** Foreign Key configuration.

Filename: **05_Pet_Table_FK.jpg**

That's it! You're done!