UNIVERSITY OF
WESTMINSTER

INFORMATICS
INSTITUTE OF
TECHNOLOGY

Module Leader MR. Kaneeka Vidanage

Lecturer MR. Shivaraam Raghu

5COSC008C DATABSE SYSTEM COURSEWORK FINAL SUBMISSION

INDIVIDUAL COURSEWORK PART A+B

Method of Submission: Submitted online via Blackboard

Group: G SE

Student ID: 2016053

UoW ID: w1673610

Student First Name: Dharshi

Student Last Name: Gunawardana

DECEMBER 10, 2018

Contents

1. Introduction

This is DATABASE SYSTEMS 1st semester 2nd course work. This basically regarding Two topics. Those are Part A and Part B. So; in the part A them contain AudioVizzion and in part B contain Futurity. I show very clear Conceptual diagram, four data dictionary tables in Part A. And I clearly show Logical diagram, wrote step-by-step guide for logical ERD, SQL code (DDL) for creating 2 tables, SQL code (DDL) for inserting record into one table, PHP code to allow user to enter data, PHP code to add record to a table and all the screenshots for above requirements. All the part B codes are in the appendices.

In this assignment they looking how to draw a conceptual diagram for given scenario and identify entity, relationship of entities, multiplicity, primary key and attributes.

Also, how to draw logical diagram for given conceptual diagram. In logical diagram how to identify primary Keys and Foreign Keys also relationship between entities, Normalization etc. Then how to Create a database using Php My Admin. And also, how to create two tables, how to insert data into table also identify primary key, foreign key, not null etc. Then how create a Web-based form and Web-based confirmation page using php. From the web-based form add data and update to database. Then retrieving data from database and show it in Web-based confirmation page also sort data.

2. Part A

2.1. Conceptual Entity Relationship Diagram for AudioVizzion

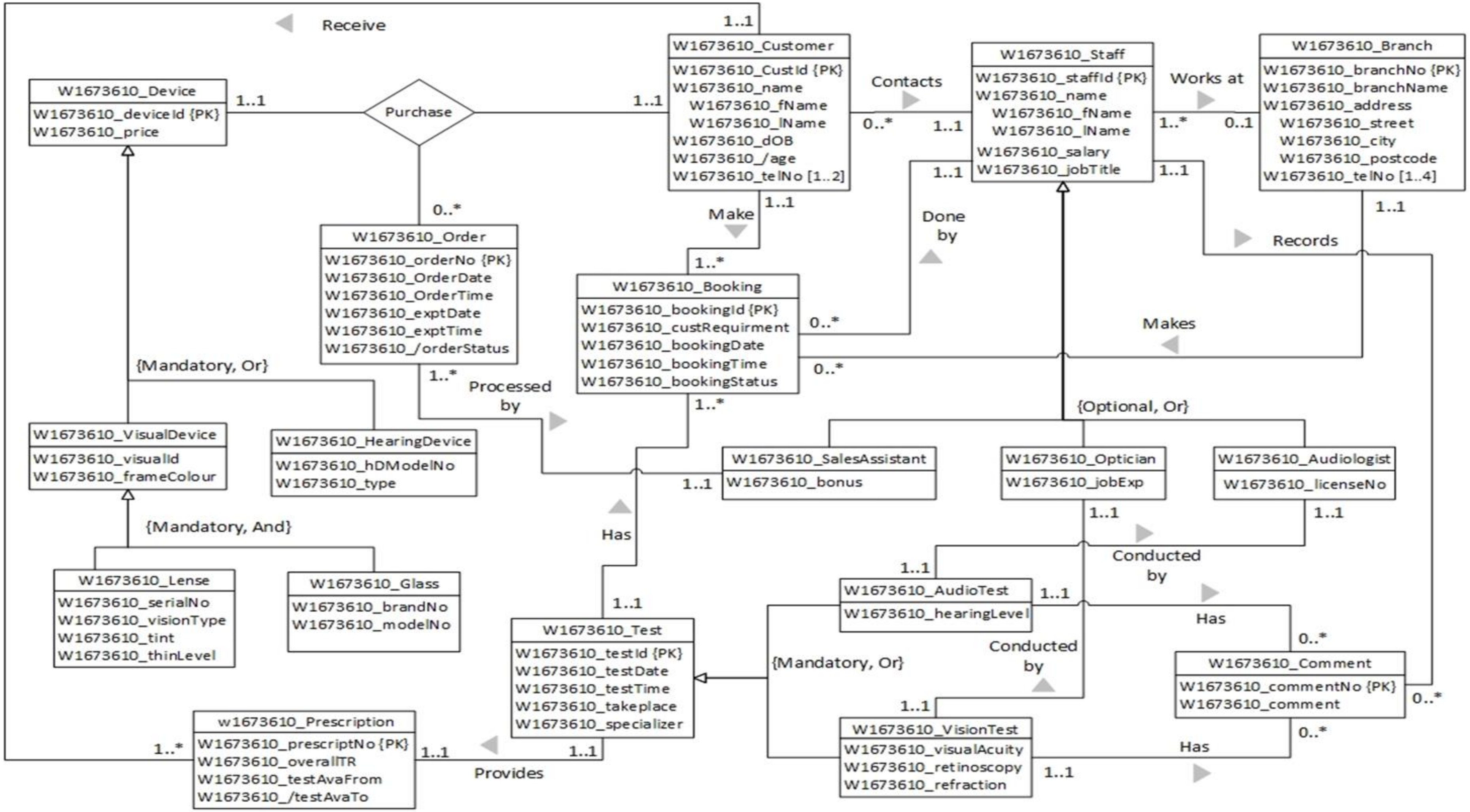


Figure 1- AudioVizzion Conceptual Diagram

2.2. Data dictionary for identify Entity

2.2.1. Identify Entity

	Entity name	Justification
1	W1673610_Branch	Branch is a business unit that provides devices and testing to the customers.
2	W1673610_Staff	Group of people are representing in the Staff (employee).
3	W1673610_SalesAssistant	Direct product supplies to the customers.
4	W1673610_ Optician	Specialized member who doing vision testing for customers.
5	W1673610_Audiologist	Specialized member who doing hearing testing for customers.
6	W1673610_Customer	A person who wants to get services and buy devices from AudioVizzion.
7	W1673610_Booking	An arrangement for getting a service.
8	W1673610_Test	To represent services that AudioVizzion offers.
9	W1673610_ AudioTest	The service to check patient hearing level.
10	W1673610_VisionTest	The service to check patient’s eye.
11	W1673610_Comment	Opinions for tests.
12	W1673610_Prescription	To represent the summery of test results.
13	W1673610_Order	The person who place an order for a device from AudioVizzion.
14	W1673610_Device	Represent products/items that AudioVizzion offers.
15	W1673610_VisualDevice	To represent visual aid/visual kit.
16	W1673610_Lens	AudioVizzion offers, to their Customer to select various type of lens for the device.
17	W1673610_Glass	AudioVizzion offers, to their Customer to select various type of glass for the device.
18	W1673610_HearingDevice	To represent hearing aid/hearing kit.

Table 1- Entities

2.2.2. Specialization/ Generalization

	General entity	Specialised entity	Justification
Staff superclass and the *Optician, Audiologist* and *SalesAssistant* are subclasses; Staff entity classified as SalesAssistant, Optician, Audiologist			
1	W1673610_Staff	W1673610_SalesAssistant	The *SalesAssistant* subclass inherits all the attributes of the *Staff* superclass (*staffId, name, salary, jobTitle*) together with those specifically associated with *SalesAssistant* subclass such as *bonus* (Connolly & Begg, 2014)
2	W1673610_Staff	W1673610_ Optician	The *Optician* subclass inherits all the attributes of the *Staff* superclass (*staffId, name, salary, jobTitle*) together with those specifically associated with *Optician* subclass as jobExp (Connolly & Begg, 2014)
3	W1673610_Staff	W1673610_Audiologist	The *Audiologist* subclass inherits all the attributes of the *Staff* superclass (*staffId, name, salary, jobTitle*) together with those specifically associated with *Audiologist* subclass as *licenseNo* (Connolly & Begg, 2014)
Test superclass and the *AudioTest, VisionTesting* are subclasses; Test entity classified as AudioTest and VisionTest			
4	W1673610_Test	W1673610_ AudioTest	The *AudioTest* subclass inherits all the attributes of the *Test* superclass (*testId, testDate, testTime, takePlace*) together with those specifically associated with *AudioTest* subclass as *hearingLevel* (Connolly & Begg, 2014)
5	W1673610_Test	W1673610_VisionTest	The *VisionTest* subclass inherits all the attributes of the *Test* superclass (*testId, testDate, testTime, takePlace*) together with those specifically associated with *VisionTest* subclass such as *visualAcuity*, *retinoscopy*, *refraction* (Connolly & Begg, 2014)
Device is a *superclass* and the *VisualDevice, HearingDevice* are subclasses; Device entity classified as VisualDevice and HearingDevice			
6	W1673610_Device	W1673610_VisualDevice	The VisualDevice subclass inherits all the attributes of the *Device* superclass (*deviceId, price*) together with those specifically associated with VisualDevice subclass such as *visualId* and *framecolour* (Connolly & Begg, 2014)
7	W1673610_Device	W1673610_HearingDevice	The HearingDevice subclass inherits all the attributes of the *Device* superclass (*deviceId, price*) together with those specifically associated with HearingDevice subclass such as *hDModelNo* and *type* (Connolly & Begg, 2014)
Glass and *Lens* are subclasses of *VisualDevice*, which is a subclass of Device; VisualDevice sub entity classified as Lens and Glass			

8	W1673610_Device	W1673610_Lens	The attributes of the *Device* superclass (*deviceId*, *price*) and the attribute of *VisualDevice* subclass (visualId, framecolour) are inherited by the *Lens* subclass, which also has its own additional attributes called *serialNo, visionType, tint, thinLevel* (Connolly & Begg, 2014)
9	W1673610_Device	W1673610_Glass	The attributes of the *Device* superclass (*deviceId*, *price*) and the attribute of *VisualDevice* subclass (visualId, framecolour) are inherited by the *Glass* subclass, which also has its own additional attributes called *brandNo, ModelNo* (Connolly & Begg, 2014)

Table 2- Specialization/ Generalization

2.3. Data dictionary for Relationships and Multiplicities

2.3.1. Binary relationship

Entity name	Multiplicity	Relationship	Multiplicity	Entity name	Justifications for the multiplicity
W1673610_Staff	1..*	Works at	0..1	W1673610_Branch	Not all Staff members *works at* a specific branch (optional participation for staff) (Connolly & Begg, 2014)
					Many Staff members *works at* one specific Branch
					One Branches *have* many Staff members
					All Branch have Staff (Connolly & Begg, 2014)
Staff and branch have Many to One relationship, (Connolly & Begg, 2015) Each Branch has one or many members of staff; Many Staff members works at zero or one Branch					
W1673610_Customer	0..*	Contacts	1..1	W1673610_Staff	Not all Staff *contacted by* a specific Customer
					Many Customers *contacts* only one Staff member
					All customers have to *contacts* (mandatory participation for Customer)
					A one specific Staff member *contacted by* many customers
Customer and staff have many to one relationship, A specific Staff member conducted by zero or many customers; Each Customers contact one Staff member					
W1673610_Customer	1..1	Make	1..*	W1673610_Booking	One Customer *make* only one Booking
					A specific Customer *make* many Bookings
					One Booking *made by* only one specific Customer
					Many Booking *made by* one Customer

Customer and booking have one to many relationships, A specific test booking makes by one customer; One Customers make one or many bookings					
W1673610 – Booking	0..*	Done by	1..1	W1673610 – Staff	Not all Staff *does* Booking (booking can be cancelling)
					Many Bookings *done by* only one Staff member
					Each member of *Staff does* many Bookings for a patient
					All Bookings are *done by* a Staff member
Booking and staff have many to one relationship, A specific Staff member do zero or many bookings; Many bookings done by one Staff member					
W1673610 – Test	1..1	Has	1..*	W1673610 – Booking	A one specific *Test has* only one *Booking*
					A one specific *Test has* many *Bookings*
					One Booking *had* only one *Test*
					Many Bookings *had* only one Test
Test and Booking have one to many relationships, A specific Booking had one Test; A specific Test has one or many bookings (Connolly & Begg, 2015)					
W1673610 – Customer	1..1	Receive	1..*	W1673610 – Prescription	One Customer *receive* only one Prescription
					One Customers *receive* many Prescription
					One Prescription *received by* only one Customer
					Many Prescriptions *receive by* only one Customer
Customer and Prescription have one to many relationships, A specific Prescription received by one customer; A specific Customer receive one or many Prescriptions					
W1673610 – Order	1..*	Processed by	1..1	W1673610 – SalesAssistant	One Order *processed by* only one SalesAssistant
					Many Orders *processed by* only one SalesAssistant
					Each SalesAssistant *process* only one Order
					Each SalesAssistant *process* many Orders
Order and Sales Assistant have many to one relationship, A specific Sales Assistant process one or many Orders; Many Orders processed by one Sales Assistant					
W1673610 – VisionTest	1..1	Conducted by	1..1	W1673610 – Optician	One Vision test *conducted by* only one Optician
					A specific Vision test *conducted by* one specific Optician
					One specific Optician *conducts* only one Vision test
					A specific Optician *conducts* one Vision test
Vision test and optician have one to one relationship, A specific optician conducts one vision test; A specific vision test conducted by one optician					

W1673610_AudioTest	1..1	Conducted by	1..1	W1673610_Audiologist	One Audio test *conducted by* only one specific Audiologist
					A specific Audio test *conducted by* one specific Audiologist
					One Audiologist *conducts* only one specific Audio test
					A specific Audiologist *conducts* one specific Audio test
Audio test and Audiologist have one to one relationship, A specific Audiologist conducts one Audio test; A specific Audio test conducted by one Audiologist					
W1673610_Staff	1..1	Records	0..*	W1673610_Comment	One Staff *records* only many comments (Connolly & Begg, 2014)
					Not all Staff *records* comments
					All Comments are *recorded* (mandatory participation for Comment)
					Many *comments recorded by* only one Staff member
Staff and Comment have one to many relationships, Each comment recorded by one Staff member; A specific staff member records zero or many comments					
W1673610_Test	1..1	Provides	1..1	W1673610_Prescription	One test *provides* only one Prescription
					One prescription *provides by* only one test (Connolly & Begg, 2014)
					A specific test *provides* one Prescription
					A specific prescription *provides by* one specific test
Test and Prescription have one to one relationship, A specific Prescription provided by one Test; A specific Test provides one Prescription					
W1673610_VisionTest	1..1	Has	0..*	W1673610_Comment	Each Vision test *has* many comments
					Not all Vision test *has* comments (optional participation for a test)
					Many comments *are in* only one Vision test
					All comments *are in* one vision Test (Connolly & Begg, 2014)
Vision test and comment have one to many relationships, Each comment had one Vision test; A specific Vision test has zero or many comments					
W1673610_AudioTest	1..1	Has	0..*	W1673610_Comment	Each Audio test *has* many comments
					Not all Audio test *has* comments (optional participation for a test)
					Many comments *are in* only one Audio test
					All comments *are in* one Audio Test (Connolly & Begg, 2014)
Audio test and comment have one to Many relationships, Each comment had one Audio test; A specific Audio test has zero or many comments					

W1673610_Branch	1..1	Makes	0..*	W1673610_Booking	Many Bookings are *made by* one Branch One Branch makes *many* Bookings All Bookings are *made* (mandatory participation for Booking) Not all Branch (member) *makes* Bookings
Branch and Booking have one to Many relationships, Each Booking mad one Branch; A specific Branch makes zero or many Bookings					

Table 3- Binary Multiplicity

2.3.2. Non-Binary relationship

Entity	Multiplicity	Relationship	Multiplicity	Entity	Justification
W1673610_Customer	1..1	Purchase	0..*	W1673610_Order	Many Orders are *purchased by* one Customer One Customer *purchase* many Orders All Orders are *purchased* Not all Customer *purchase* orders
W1673610_Customer Order	1..1	Purchase	1..1	W1673610_Device	One Device *purchased by* only one Customer A specific Customer *purchase* one Device A specific Device *purchased by* one Customer One Customer *purchase* only one Device
A specific customer makes zero or many Orders purchase for only one device at a time					

Table 4- Non-Binary Multiplicity

Customer make an order to purchase a device. So, purchased Order responsible by Sales Assistance at a Branch. A customer makes one order; device is allocated to that order. A customer makes many orders; devices are separately allocated to those orders (by deviceId and orderNo). For an order there have only one device.

2.3.3. Constraints on Specialization/ Generalization

The relationship between subclass and super class (1:1)

- Specialization/Generalization the relationship between Staff with SalesAssistant, Optician, Audiologist

Staff entity is optional and disjoint {optional, or}, as not all members of staff are SalesAssistant or Optician or Audiologist, and also a single member of staff cannot be overlap with SalesAssistant and Optician and Audiologist (Connolly & Begg, 2014).

- Specialization/Generalization the relationship between Test with AudioTest and VisionTest.

Test entity is mandatory and disjoint {mandatory, or}, as Test must be either AudioTest or VisionTest but cannot be both (Connolly & Begg, 2014). [the same booking id can't do two tests]

- Specialization/Generalization the relationship between Device with VisualDevice, HearingDevice, and Lens, Glass.

Device entity is mandatory and disjoint {mandatory, or}, as Device must be Visual Device (as visual device non-disjoint with lens and glasses {mandatory, and} can be both lens and glass) or Hearing Device but cannot be both (Connolly & Begg, 2014).

2.4. Data dictionary for attributes and primary keys

For some entities I use primary keys, Multi-valued attributes, derived attributes.

	Entity name	Attributes for this entity (include PK)	Justification
1	W1673610_Branch	W1673610_branchNo {PK}	Uniquely identities a branch
		W1673610_branchName	Name of the branch
		W1673610_address	
		W1673610_street	Street of the branch
		W1673610_city	City of the branch
		W1673610_postcode	Postcode of the branch
		W1673610_telNo [1..4]	telephone numbers for branch
2	W1673610_Staff	W1673610_staffId {PK}	Uniquely identities the employee
		W1673610_name	
		W1673610_fName	First name of the employee in the branch
		W1673610_lname	Last name of the employee in the branch
		W1673610_salary	Salary of the employee
		W1673610_jobTitle	The position of the employee in the branch
3	W1673610_SalesAssistant	W1673610_bonus	Bonus for a sales assistant
4	W1673610_Optician	W1673610_jobExp	Job experience for an optician
5	W1673610_Audiologist	W1673610_licenseNo	government License number for an audiologist
6	W1673610_Customer	W1673610_custId {PK}	Uniquely identities a patient/customer
		W1673610_name	
		W1673610_fName	First name of a customer
		W1673610_lName	Last name of a customer
		W1673610_dOB	Date of birth for a customer
		W1673610_/age	Customer age (calculate current date from dob)
		W1673610_telNo [1..2]	Telephone number for a customer

7	W1673610_Booking	W1673610_bookingId {PK}	Uniquely identities a patient booking for a test
		W1673610_custRequirment	Customer requirements; what actually customer wants to test.
		W1673610_bookingDate	Test booking date (customer contact date)
		W1673610_bookingTime	Test booking time (customer contact time)
		W1673610_bookingStatus	Booking confirmation (confirmation- true/ false)
8	W1673610_Test	W1673610_testId {PK}	Uniquely identities a test
		W1673610_testDate	Staff member given date for the patient to the request test
		W1673610_testTime	Staff member given time for the patient to the request test
		W1673610_takePlace	Staff member given place for the patient to the request test
		W1673610_specializer	Staff member given specialised person for the patient to the request test
9	W1673610_AudioTest	W1673610_hearingLevel	Measures the patient’s hearing level
10	W1673610_VisionTest	W1673610_visualAcuity	Measures patient visual Acuity
		W1673610_retinoscopy	Measures patient eye retinoscopy
		W1673610_refraction	Measures patient eye refraction
11	W1673610_Comment	W1673610_commentNo {PK}	Each comment uniquely identifies by comment number
		W1673610_comment	Opinion for an individual sub-test
12	W1673610_Prescription	W1673610_prescriptNo {PK}	Uniquely identities an overall test result
		W1673610_overallTR	Detailed summary of the test results
		W1673610_testAvaFrom	Date from which it is available
		W1673610_/testAvaTo	date until when it is available (add years to testAvaFrom)
13	W1673610_Order	W1673610_orderNo {PK}	Uniquely identities a customer ordering
		W1673610_orderDate	Customer purchase date
		W1673610_orderTime	Customer purchase time
		W1673610_exptDate	Expected collecting date
		W1673610_exptTime	Expected collecting time
		W1673610_/status	Status; for customers collected product yet (From exptDate and actual date)
14	W1673610_Device	W1673610_deviceId {PK}	Uniquely identities a purchased device
		W1673610_price	Price of the device
15	W1673610_VisualDevice	W1673610_visualId	Owns visual identification number
		W1673610_frameColour	Device frame colour

16	W1673610 _ Lens	W1673610_serialNo	Own serial number for a lens
		W1673610_visionType	Own may Types of vision
		W1673610_tint	Tint of lens
		W1673610_thinLevel	Level of thinness
17	W1673610 _ Glass	W1673610_brandNo	Own Brand number for a glass
		W1673610_modelNo	Own model number for a glass
18	W1673610 _ HearingDevice	W1673610_hDModelNo	Own model number for a hearing device
		W1673610_type	type that customer requested hearing device

Table 5- Attributes

2.5. Reference

Connolly, T. & Begg, C., 2014. *Database System.* fourth ed. India: Pearson.

Connolly, T. & Begg, C., 2015. *Database Systems,* University of the west of Scotland: Person.

2.6. Assumptions

I. In this company there have large number of branches. So, we need to separately identify those branches.

II. All the staff members are involved in to AudioVizzion company. And in that there have many branches to sell devices and services. And all the staff members are not involved into a specific branch.

III. A branch has 1 - 4 contact numbers for customers to contact with the staff. (min-1, max-4)

IV. Every branch are offers services and tests, So I assume a customer first contact a member of staff at a branch to make a booking for a test (most convenient: through the staff member).

V. Comments are given by a staff member that I assume that comments are represent that individual sub testing (audio, vision testing) “comments” such as visual acuity: 20/20

VI. As I have assumed, In the Prescription give a detailed summary of the published results (with the comments). Such as Visual acuity: Normal (20/20)

VII. A customer can make two appointments such as vision and hearing tests. But Staff member who is confirm those bookings.

VIII. For a test there have only one prescription. And a customer can get one or many prescriptions because if a customer did two tests so, then customer can get two prescriptions.

IX. At the end of a test the staff member records no. of comments so I assume comments are not essential for every sub test.

X. At a time one booking has only one test. Once the test end then after that other test comes one by one. If customer books two tests, then one after the other test. (at a time, a customer can appear a one test)

XI. A customer can make an order, or do only a test. Because if customer don’t have problem on hear, or eye then they don’t want to order a device.

XII. I assume a person can purchase many orders. But those many orders can be represent one device at a time.

3. Part B

3.1. Logical Entity Relationship Diagram for Futurity

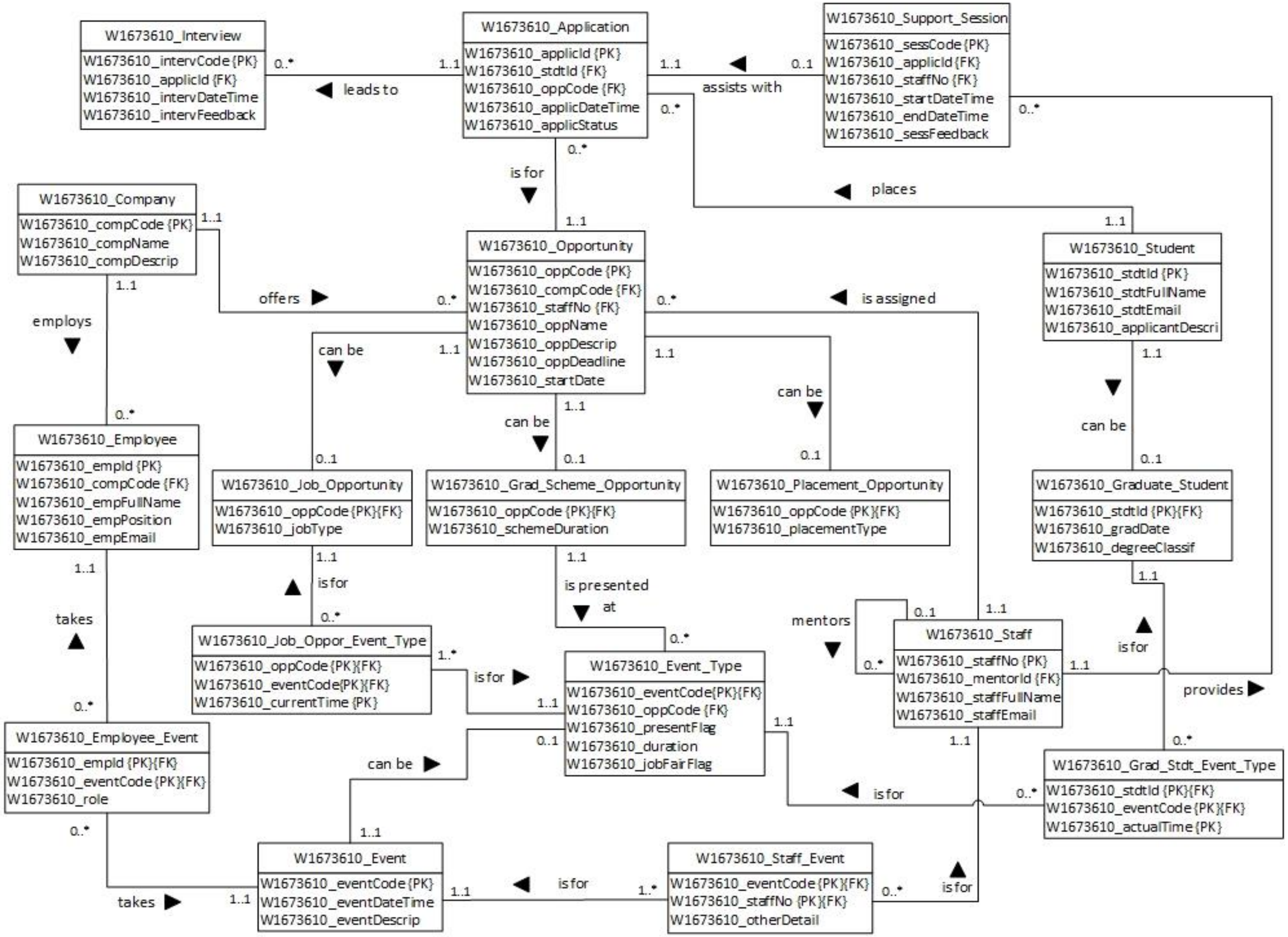


Figure 2- Futurity Logical Diagram

3.2. Step-By-Step Guide for Logical ERD

- ‘One to one relationship holds in **Application** and **Support_Session** and this Support_Session table on the ‘optional’ side. So; Primary Key of the Application goes to Support_Session as Foreign Key.

- ‘One to many’ relationships hold in **Application** and **Interview**. So; Primary Key of the Application goes to Interview as Foreign Key.

- ‘One to many’ relationships hold in **Student** and **Application**. So; Primary Key of the Student goes to Application as Foreign Key.

- ‘One to many’ relationships hold in **Opportunity** and **Application**. So; Primary Key of the Opportunity goes to Application as Foreign Key.

- ‘One to many’ relationships hold in **Staff** and **Support_Session**. So; Primary Key of the Staff goes to Support_Session as Foreign Key.

- ‘One to many’ relationships hold in **Staff** and **Opportunity**. So; Primary Key of the Staff goes to Opportunity as Foreign Key.

- ‘One to many’ relationships hold in **Company** and **Employee**. So; Primary Key of the Company goes to Employee as Foreign Key.

- ‘One to many’ relationships hold in **Company** and **Opportunity**. So; Primary Key of the Company goes to Opportunity as Foreign Key.

- ‘Optional, Or’ Relationship in **Opportunity, Placement, Job, Grad_Scheme.** So; they connect with ‘One to one’ relationship with child side ‘Optional’. Renamed sub entities. Also, Primary Key of the Opportunity (Parent Table) goes to those three **Placement_ Opportunity**, **Job_ Opportunity**, **Grad_Scheme_ Opportunity** tables (Child Tables) as Primary Key and also Foreign Key.

- ‘Optional, And’ Relationship in **Event, Presentation, Job_Fair.** So; they connect with ‘One to one’ relationship with child side ‘Optional’. Both Presentation, Job_Fair merge together and create **Event_Type** (child table). Also, Primary Key of the Event (Parent Table) goes to those Event_Type table as Primary Key and also Foreign Key. Use flags to differentiate between records So I use presentFlag and jobFairFlag for that.

- ‘Optional’ Relationship with in **Student** and **Graduate**. So; they connect with ‘One to one’ relationship with child side ‘Optional’. Renamed sub entity as **Graduate_Student.** Also, Primary Key of the Student (Parent Table) goes to Graduate_Student table (Child Tables) as Primary Key and also Foreign Key.

- Recursive Relationship with **Staff** and **Mentor.** So; mentor id goes to staff table as Foreign Key.

- ‘Many to Many’ Relationships hold in **Employee** and **Event.** So; with those parent tables and newly create merged table called **Employee_Event** that it holds both employee and Event table’s Primary keys and also those primary keys became Foreign keys.

- ‘Many to Many’ Relationships hold in **Staff** and **Event.** So; with those parent tables and newly create merged table called **Staff_Event** that it holds both Staff and Event table’s Primary keys and also those primary keys became Foreign keys.

- ‘Many to Many’ Relationships hold in **Graduate** and **Presentation.** As before those two tables became **Graduate_Student** and **Event_Type.** So; with those parent tables and newly create merged table called **Graduate_Stdt _Event_Type** that it holds both Graduate_Student and Event_Type table’s Primary keys and also those primary keys became Foreign keys. Also; actualTime act as Composite Primary Key.

- ‘Many to Many’ Relationships hold in **Job** and **Job_Fair.** As before those two tables became **Job_ Opportunity** and **Event_Type.** So; with those parent tables and newly create merged table called **Job_Oppor_Event_Type** that it holds both Job_ Opportunity and Event_Type table’s Primary keys and also those primary keys became Foreign keys. Also; currentTime act as Composite Primary Key.

- ‘One to many’ relationships hold in **Grad_Scheme** and **Presentation**. As before those two tables became **Grad_Scheme_ Opportunity** and **Event_Type.** So; Primary Key of the **Grad_Scheme_ Opportunity** goes to **Event_Type** as Foreign Key.

3.2.1. Relational schema

- Company (compCode {PK}, compName, compDescrip)

- Employee (empId {PK}, compCode {FK}, empFullName, empPosition, empEmail)

- Event (eventCode {PK}, eventDateTime, eventDescrip)

- Student (stdId {PK}, stdFullName, stdEmail,applicantDescri)

- Staff (staffNo {PK}, metorStaffId {FK}, staffFullName, staffEmail)

- Opportunity (oppcode {PK}, compCode {FK}, staffNo {FK}, oppName, oppDescrip, startDate)

- Application (applicId {PK}, stdId {FK}, opcode {FK}, applicDateTime,applicStatus)

- Interview (intervCode {PK}, applicId {FK}, intervDateTime, intervFeedback)

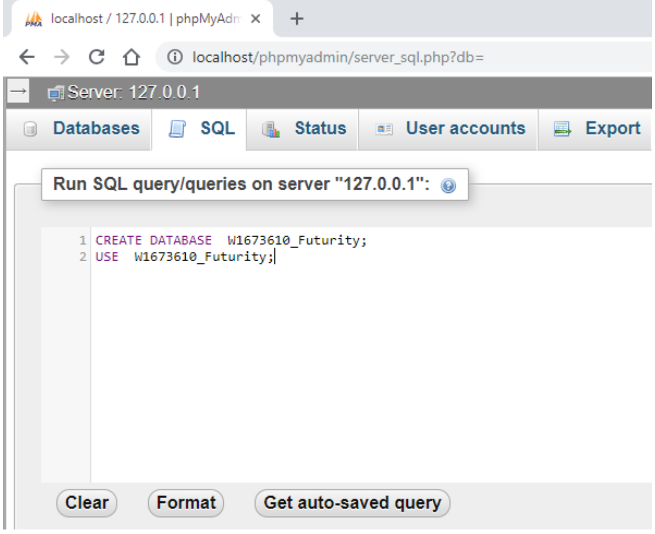
- Support Session (seesCode {PK}, applicId {FK}, staffNo {FK}, startDateTime, endDateTime, sessFeedback)

- Job Opportunity (oppcode {PK} {FK}, jobType)

- Grad Scheme Opportunity (oppcode {PK} {FK}, schemeDuration)
- Placement Opportunity (oppcode {PK} {FK}, placementType)
- Event Type (eventCode {PK} {FK}, oppcode {FK}, presentFlag, duration, JobFairFlag)
- Employee Event (empId {PK} {FK}, eventCode {PK} {FK}, role)
- Staff Event (eventCode {PK} {FK}, staffNo {PK} {FK}, otherDetail)
- Job Opportunity Event Type ((oppcode {FK}, eventCode {FK}, currentTime) {PK})
- Graduate Student (stdId {PK} {FK}, gradate, degreeClassif)
- Graduate Student Event Type ((stdId {FK}, eventCode {FK}, actualTime) {PK})

3.3. SQL code (DDL) for creating 2 tables and screenshot

SQL Statement for Creates a database called "Futurity".

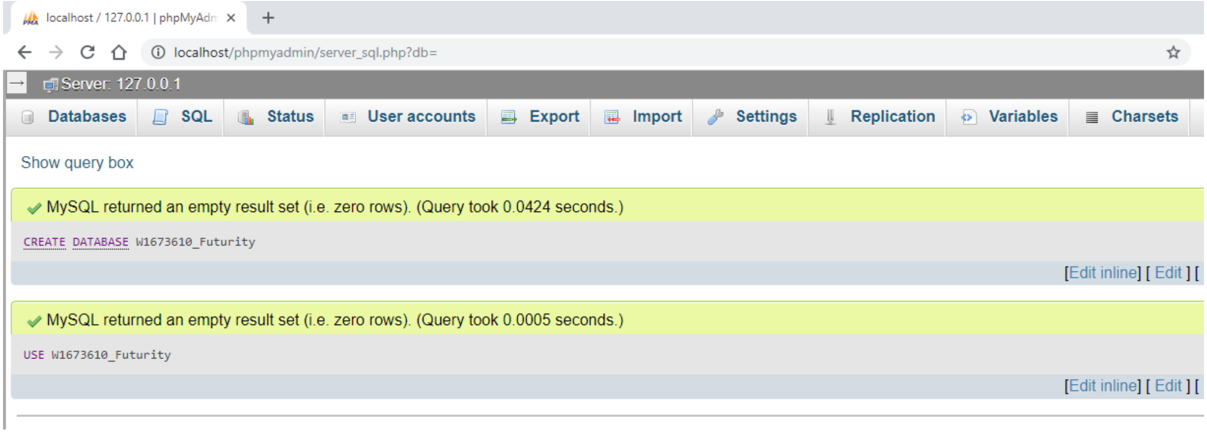


Run SQL query/queries on server "127.0.0.1":

```
CREATE DATABASE  W1673610_Futurity;
USE  W1673610_Futurity;
```

Figure 3- Create Database

Execute the SQL statement above.



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0424 seconds.)

```
CREATE DATABASE W1673610_Futurity
```

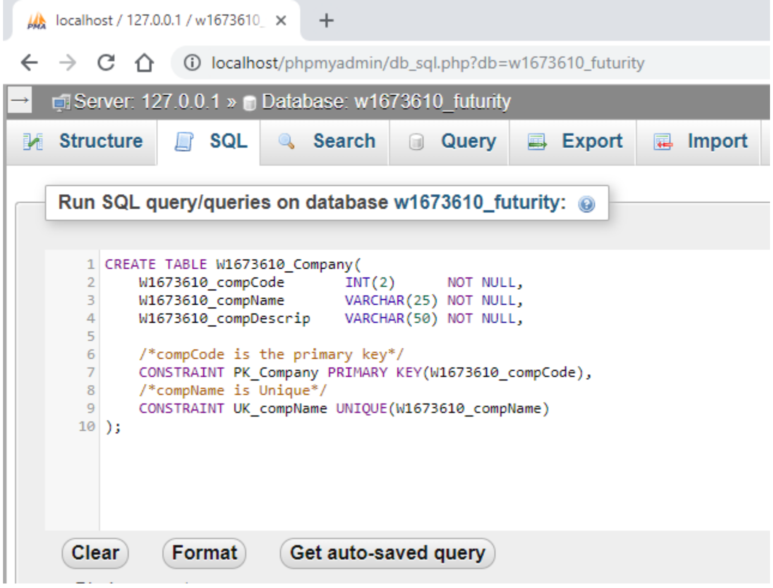
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```
USE W1673610_Futurity
```

Figure 4

3.3.1. Company Table

SQL Statement for Create Company Table

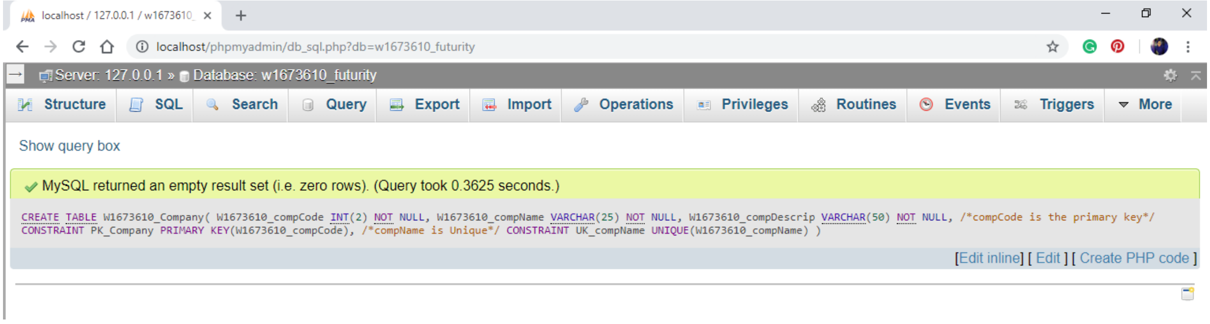

localhost / 127.0.0.1 / w1673610
localhost/phpmyadmin/db_sql.php?db=w1673610_futurity
Server: 127.0.0.1 » Database: w1673610_futurity
Structure
SQL
Search
Query
Export
Import
Run SQL query/queries on database w1673610_futurity:
Clear
Format
Get auto-saved query

```
CREATE TABLE W1673610_Company(
    W1673610_compCode       INT(2)      NOT NULL,
    W1673610_compName       VARCHAR(25) NOT NULL,
    W1673610_compDescrip    VARCHAR(50) NOT NULL,

    /*compCode is the primary key*/
    CONSTRAINT PK_Company PRIMARY KEY(W1673610_compCode),
    /*compName is Unique*/
    CONSTRAINT UK_compName UNIQUE(W1673610_compName)
);
```

Figure 5- Company Table

Execute the SQL statement above.


localhost / 127.0.0.1 / w1673610
localhost/phpmyadmin/db_sql.php?db=w1673610_futurity
Server: 127.0.0.1 » Database: w1673610_futurity
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
More
Show query box
[Edit inline] [Edit] [Create PHP code]

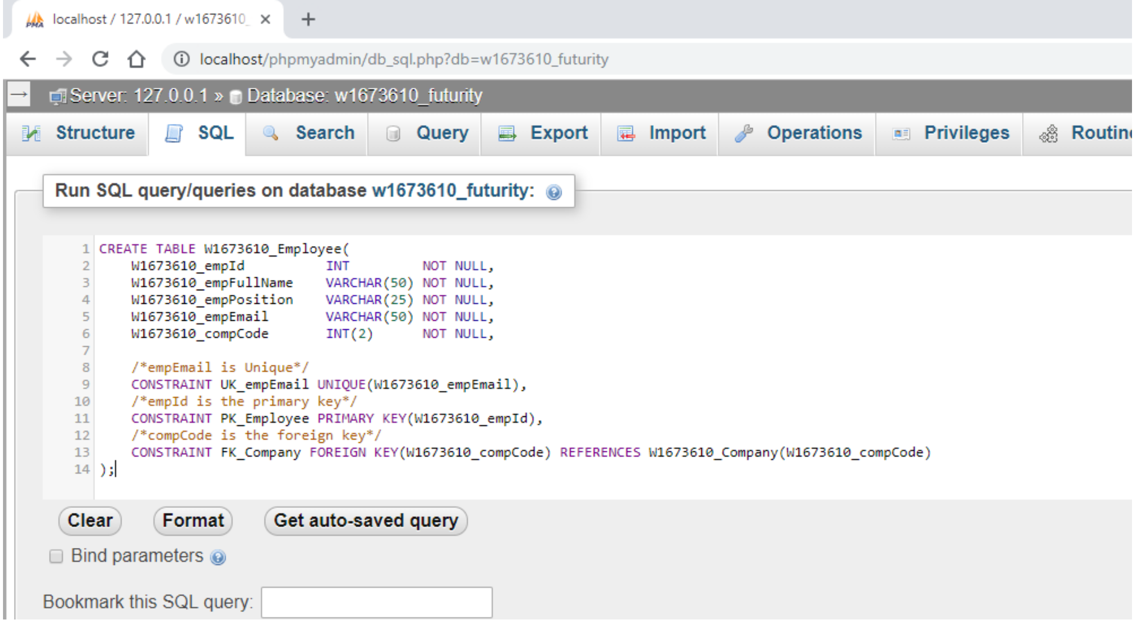
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.3625 seconds.)

```
CREATE TABLE W1673610_Company( W1673610_compCode INT(2) NOT NULL, W1673610_compName VARCHAR(25) NOT NULL, W1673610_compDescrip VARCHAR(50) NOT NULL, /*compCode is the primary key*/
CONSTRAINT PK_Company PRIMARY KEY(W1673610_compCode), /*compName is Unique*/ CONSTRAINT UK_compName UNIQUE(W1673610_compName) )
```

Figure 6

3.3.2. Employee Table

SQL Statement for Create Employee Table.

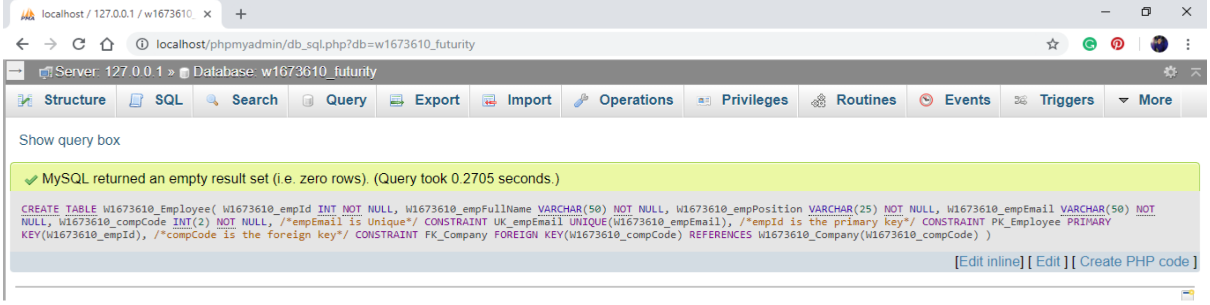

localhost / 127.0.0.1 / w1673610
localhost/phpmyadmin/db_sql.php?db=w1673610_futurity
Server: 127.0.0.1 » Database: w1673610_futurity
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routin
Run SQL query/queries on database w1673610_futurity:
Clear
Format
Get auto-saved query
□ Bind parameters
Bookmark this SQL query:

```
CREATE TABLE W1673610_Employee(
    W1673610_empId          INT         NOT NULL,
    W1673610_empFullName    VARCHAR(50) NOT NULL,
    W1673610_empPosition    VARCHAR(25) NOT NULL,
    W1673610_empEmail       VARCHAR(50) NOT NULL,
    W1673610_compCode       INT(2)      NOT NULL,

    /*empEmail is Unique*/
    CONSTRAINT UK_empEmail UNIQUE(W1673610_empEmail),
    /*empId is the primary key*/
    CONSTRAINT PK_Employee PRIMARY KEY(W1673610_empId),
    /*compCode is the foreign key*/
    CONSTRAINT FK_Company FOREIGN KEY(W1673610_compCode) REFERENCES W1673610_Company(W1673610_compCode)
);
```

Figure 7- Employee Table

Execute the SQL statement above.


localhost / 127.0.0.1 / w1673610
localhost/phpmyadmin/db_sql.php?db=w1673610_futurity
Server: 127.0.0.1 » Database: w1673610_futurity
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
More
Show query box

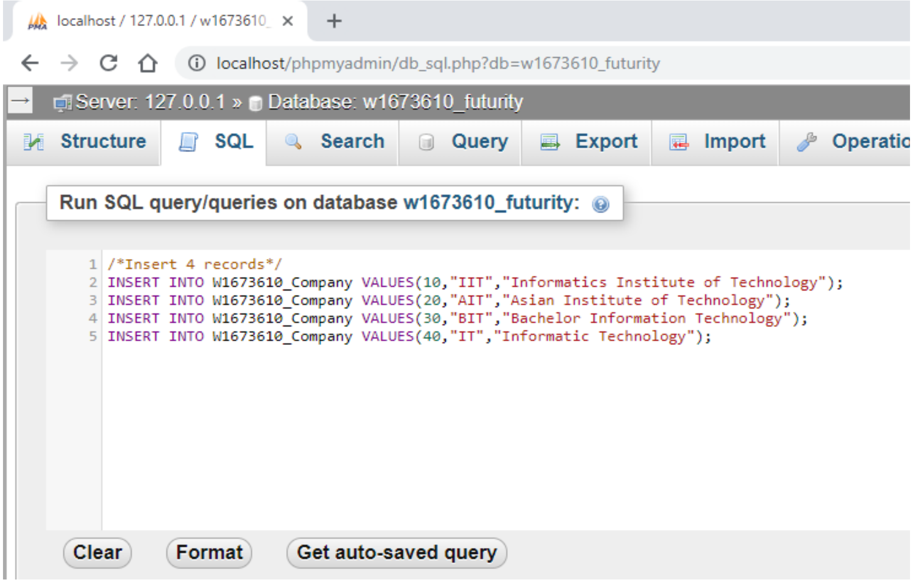
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.2705 seconds.)

```
CREATE TABLE W1673610_Employee( W1673610_empId INT NOT NULL, W1673610_empFullName VARCHAR(50) NOT NULL, W1673610_empPosition VARCHAR(25) NOT NULL, W1673610_empEmail VARCHAR(50) NOT NULL, W1673610_compCode INT(2) NOT NULL, /*empEmail is Unique*/ CONSTRAINT UK_empEmail UNIQUE(W1673610_empEmail), /*empId is the primary key*/ CONSTRAINT PK_Employee PRIMARY KEY(W1673610_empId), /*compCode is the foreign key*/ CONSTRAINT FK_Company FOREIGN KEY(W1673610_compCode) REFERENCES W1673610_Company(W1673610_compCode) )
```

Figure 8

3.4. SQL code for inserting record into Company Table and screenshot

SQL Statement for Insert records to Company

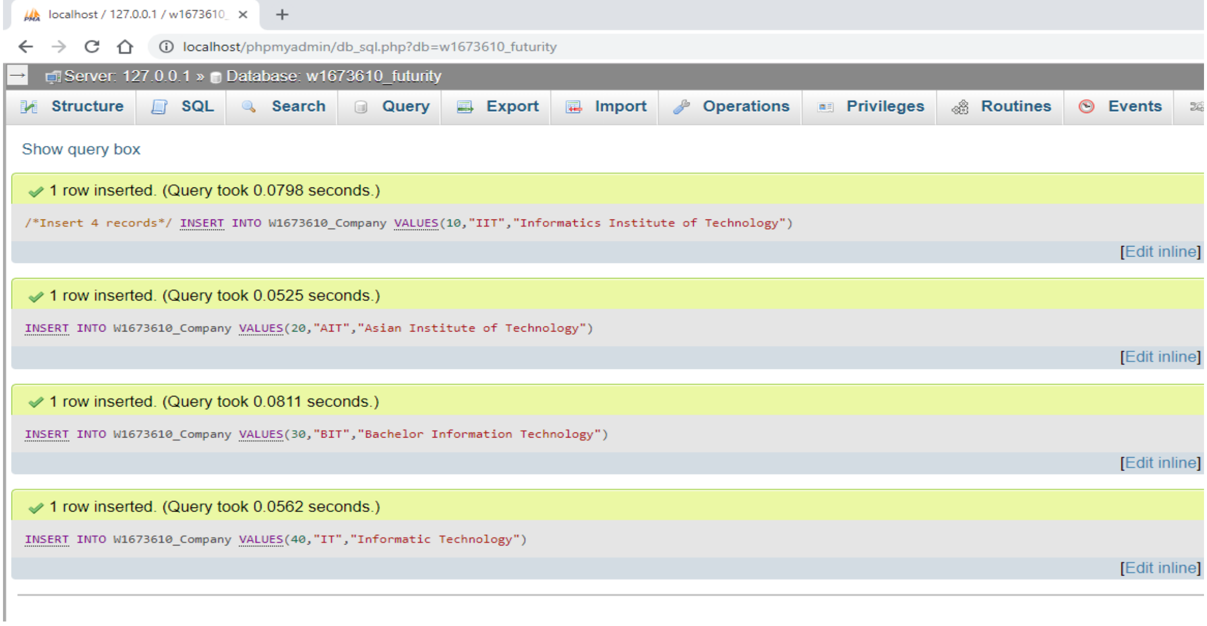

localhost / 127.0.0.1 / w1673610
localhost/phpmyadmin/db_sql.php?db=w1673610_futurity
Server: 127.0.0.1 » Database: w1673610_futurity
Structure
SQL
Search
Query
Export
Import
Operatio
Run SQL query/queries on database w1673610_futurity:

```
/*Insert 4 records*/
INSERT INTO W1673610_Company VALUES(10,"IIT","Informatics Institute of Technology");
INSERT INTO W1673610_Company VALUES(20,"AIT","Asian Institute of Technology");
INSERT INTO W1673610_Company VALUES(30,"BIT","Bachelor Information Technology");
INSERT INTO W1673610_Company VALUES(40,"IT","Informatic Technology");
```

Clear
Format
Get auto-saved query

Figure 9- Insert data into Company Table

Execute the SQL statement above.


localhost / 127.0.0.1 / w1673610
localhost/phpmyadmin/db_sql.php?db=w1673610_futurity
Server: 127.0.0.1 » Database: w1673610_futurity
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Show query box

✓ 1 row inserted. (Query took 0.0798 seconds.)

```
/*Insert 4 records*/ INSERT INTO W1673610_Company VALUES(10,"IIT","Informatics Institute of Technology")
```

✓ 1 row inserted. (Query took 0.0525 seconds.)

```
INSERT INTO W1673610_Company VALUES(20,"AIT","Asian Institute of Technology")
```

✓ 1 row inserted. (Query took 0.0811 seconds.)

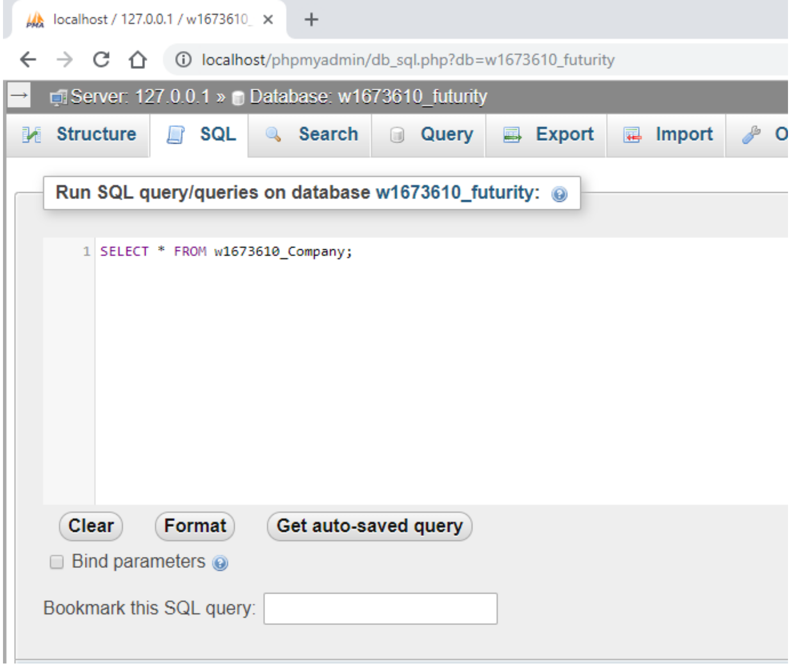
```
INSERT INTO W1673610_Company VALUES(30,"BIT","Bachelor Information Technology")
```

✓ 1 row inserted. (Query took 0.0562 seconds.)

```
INSERT INTO W1673610_Company VALUES(40,"IT","Informatic Technology")
```

Figure 10

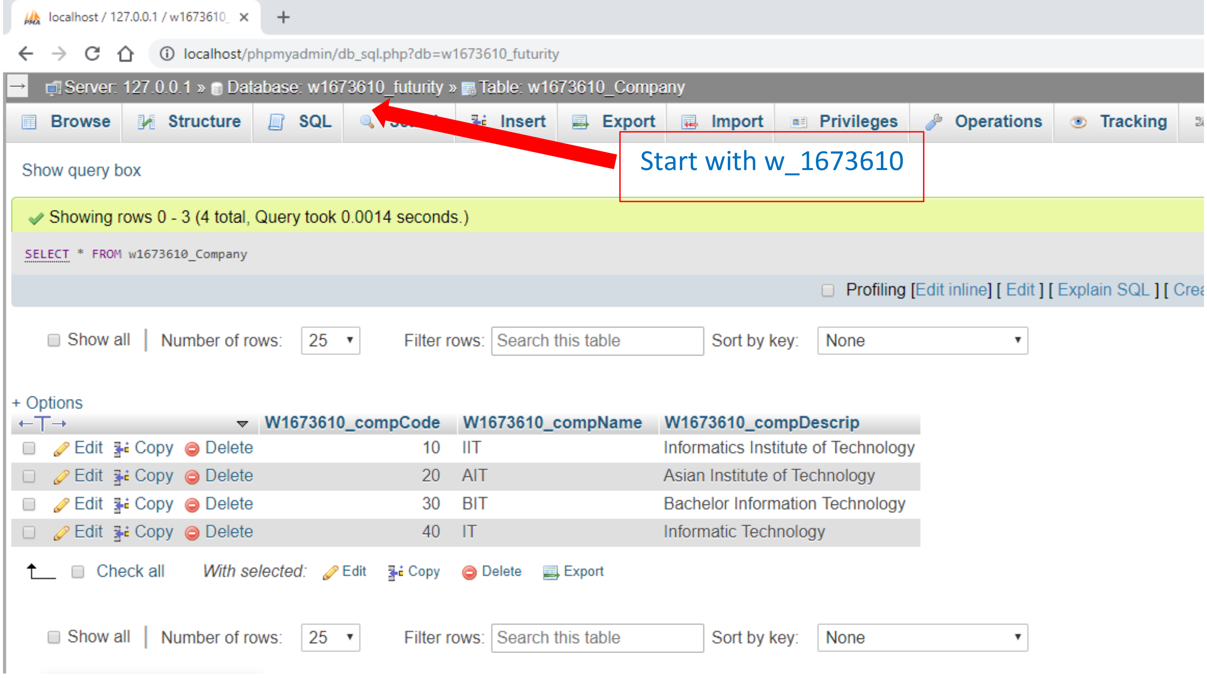
SQL statement selects the " compCode ", “compName”, “compDescrip” columns from the "Company" table


localhost / 127.0.0.1 / w1673610
localhost/phpmyadmin/db_sql.php?db=w1673610_futurity
Server: 127.0.0.1 » Database: w1673610_futurity
Structure
SQL
Search
Query
Export
Import
Run SQL query/queries on database w1673610_futurity:
Clear
Format
Get auto-saved query
Bind parameters
Bookmark this SQL query:

```
SELECT * FROM w1673610_Company;
```

Figure 11- result-set

Execute the SQL statement above.


localhost / 127.0.0.1 / w1673610
localhost/phpmyadmin/db_sql.php?db=w1673610_futurity
Server: 127.0.0.1 » Database: w1673610_futurity » Table: w1673610_Company
Browse
Structure
SQL
Insert
Export
Import
Privileges
Operations
Tracking
Start with w_1673610
Show query box
Profiling [Edit inline] [Edit] [Explain SQL] [Crea
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options
Check all
With selected:
Edit
Copy
Delete
Export

✓ Showing rows 0 - 3 (4 total, Query took 0.0014 seconds.)

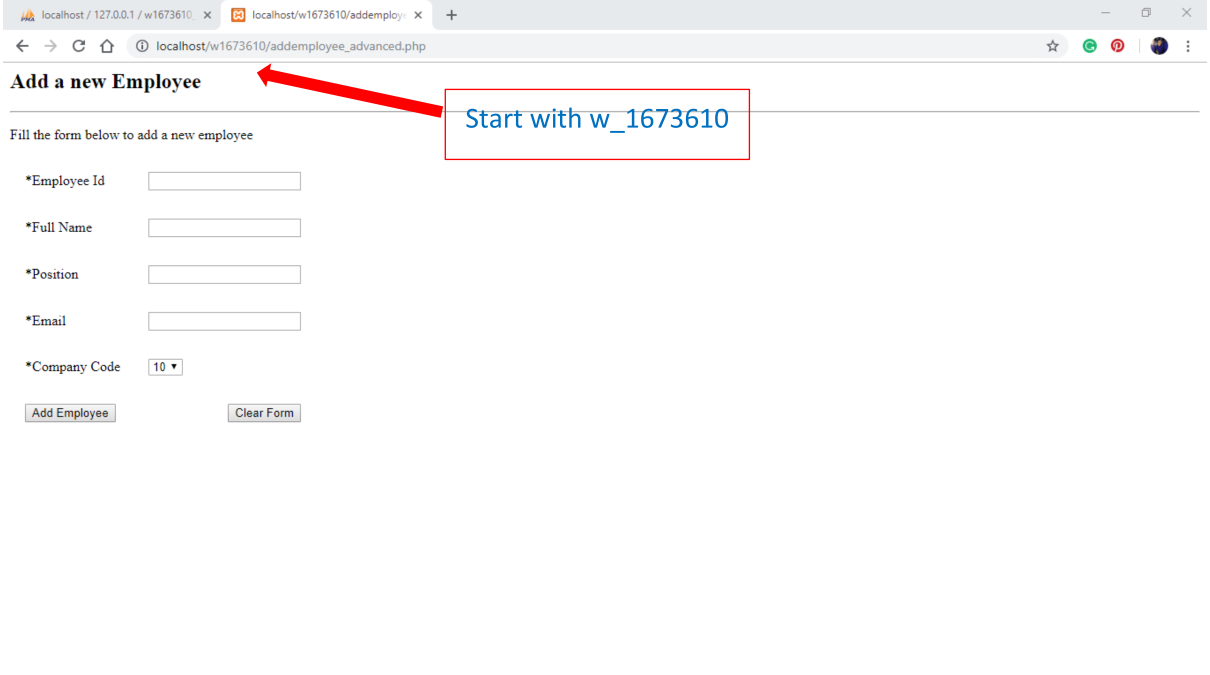
```
SELECT * FROM w1673610_Company
```

	W1673610_compCode	W1673610_compName	W1673610_compDescrip
□ Edit Copy Delete	10	IIT	Informatics Institute of Technology
□ Edit Copy Delete	20	AIT	Asian Institute of Technology
□ Edit Copy Delete	30	BIT	Bachelor Information Technology
□ Edit Copy Delete	40	IT	Informatic Technology

Figure 12- Inserted data in Company Table

3.5. PHP code to add and retrieve data into and from table

Advanced version: Drop down list for company code.


localhost / 127.0.0.1 / w1673610
localhost/w1673610/addemploye
localhost/w1673610/addemployee_advanced.php

Add a new Employee

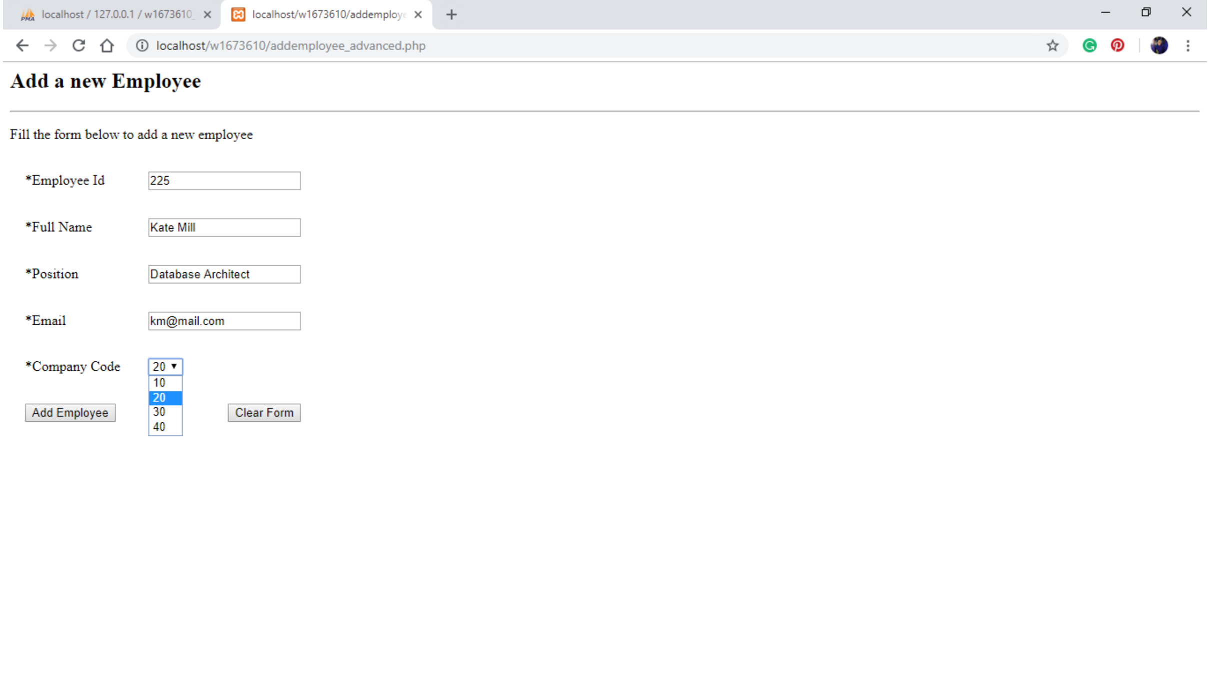
Start with w_1673610

Fill the form below to add a new employee

*Employee Id	
*Full Name	
*Position	
*Email	
*Company Code	10 ▾

Add Employee Clear Form

Figure 13- addemployee_advanced.php


localhost / 127.0.0.1 / w1673610
localhost/w1673610/addemploye
localhost/w1673610/addemployee_advanced.php

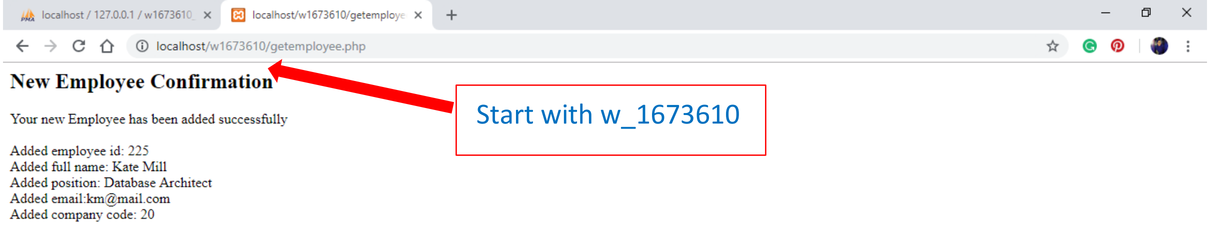
Add a new Employee

Fill the form below to add a new employee

*Employee Id	225
*Full Name	Kate Mill
*Position	Database Architect
*Email	km@mail.com
*Company Code	20 ▾ 10 20 30 40

Add Employee Clear Form

Figure 14- Drop-down option for company code


localhost / 127.0.0.1 / w1673610
localhost/w1673610/getemploye
localhost/w1673610/getemployee.php

New Employee Confirmation

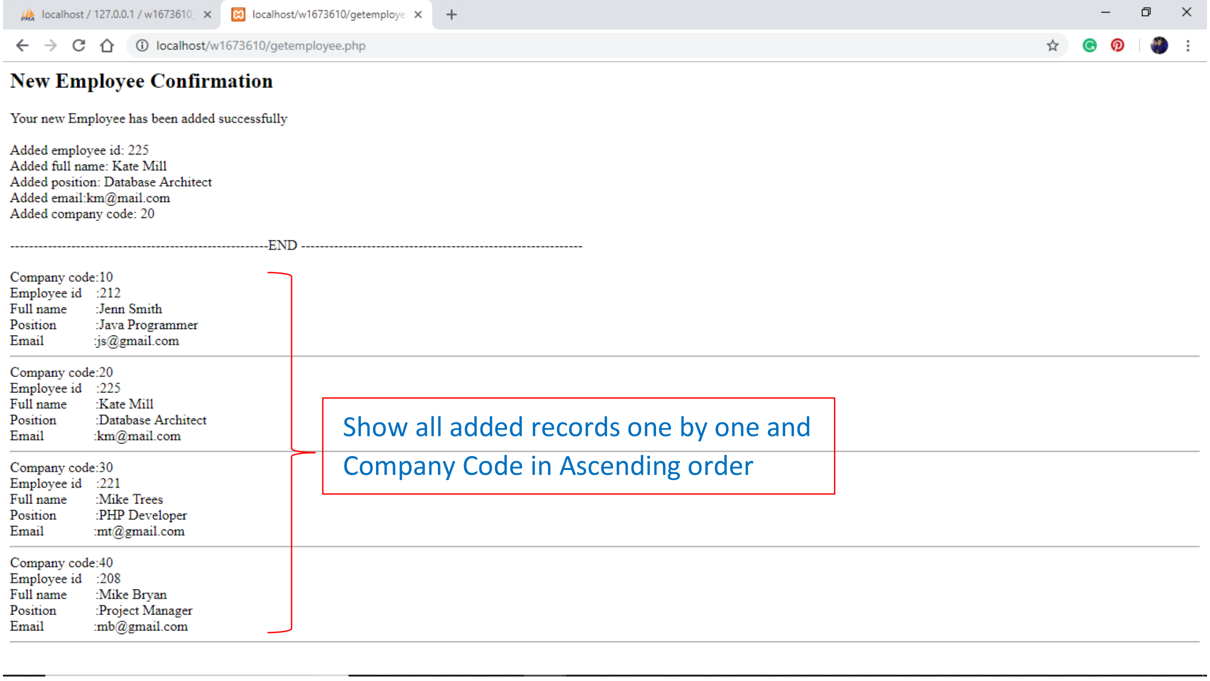
Start with w_1673610

Your new Employee has been added successfully

Added employee id: 225
Added full name: Kate Mill
Added position: Database Architect
Added email:km@mail.com
Added company code: 20

Figure 15- getemployee.php

Sort the company code in ascending order. Show all Employee records are in SQL.


localhost / 127.0.0.1 / w1673610
localhost/w1673610/getemploye
localhost/w1673610/getemployee.php

New Employee Confirmation

Your new Employee has been added successfully

Added employee id: 225
Added full name: Kate Mill
Added position: Database Architect
Added email:km@mail.com
Added company code: 20

--END --

Company code:10
Employee id :212
Full name :Jenn Smith
Position :Java Programmer
Email :js@gmail.com

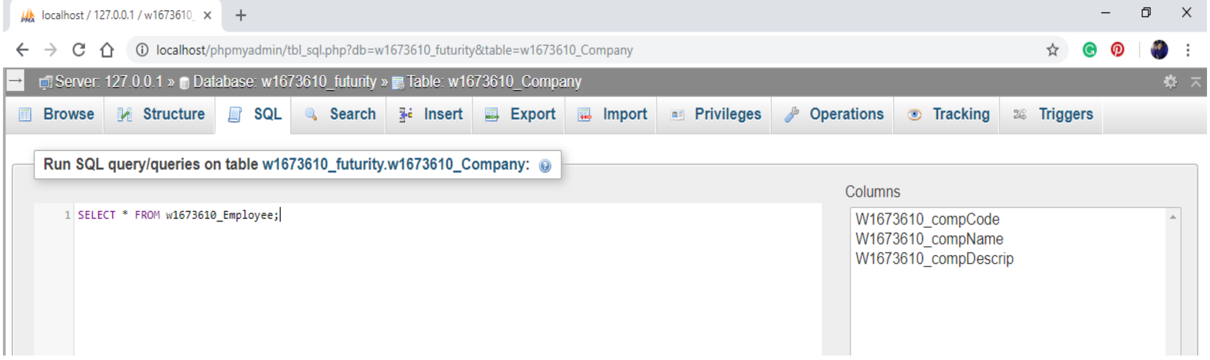
Company code:20
Employee id :225
Full name :Kate Mill
Position :Database Architect
Email :km@mail.com

Company code:30
Employee id :221
Full name :Mike Trees
Position :PHP Developer
Email :mt@gmail.com

Company code:40
Employee id :208
Full name :Mike Bryan
Position :Project Manager
Email :mb@gmail.com

Show all added records one by one and Company Code in Ascending order

Figure 16- Show all records by ascending order

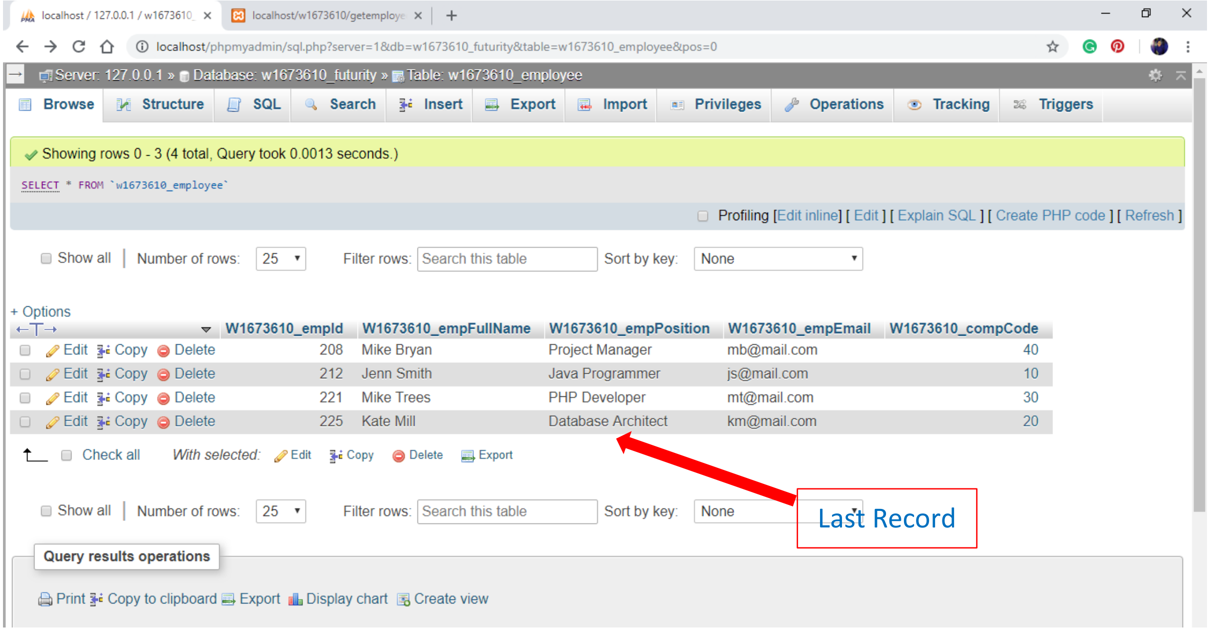

localhost / 127.0.0.1 / w1673610
localhost/phpmyadmin/tbl_sql.php?db=w1673610_futurity&table=w1673610_Company
Server: 127.0.0.1 » Database: w1673610_futurity » Table: w1673610_Company
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers

Run SQL query/queries on table w1673610_futurity.w1673610_Company:

```
SELECT * FROM w1673610_Employee;
```

Columns
W1673610_compCode
W1673610_compName
W1673610_compDescrip

Figure 17- result-set


localhost / 127.0.0.1 / w1673610
localhost/w1673610/getemploye
localhost/phpmyadmin/sql.php?server=1&db=w1673610_futurity&table=w1673610_employee&pos=0
Server: 127.0.0.1 » Database: w1673610_futurity » Table: w1673610_employee
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers

✓ Showing rows 0 - 3 (4 total, Query took 0.0013 seconds.)

```
SELECT * FROM `w1673610_employee`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	W1673610_empId	W1673610_empFullName	W1673610_empPosition	W1673610_empEmail	W1673610_compCode
□ Edit Copy Delete	208	Mike Bryan	Project Manager	mb@mail.com	40
□ Edit Copy Delete	212	Jenn Smith	Java Programmer	js@mail.com	10
□ Edit Copy Delete	221	Mike Trees	PHP Developer	mt@mail.com	30
□ Edit Copy Delete	225	Kate Mill	Database Architect	km@mail.com	20

□ Check all With selected: Edit Copy Delete Export

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

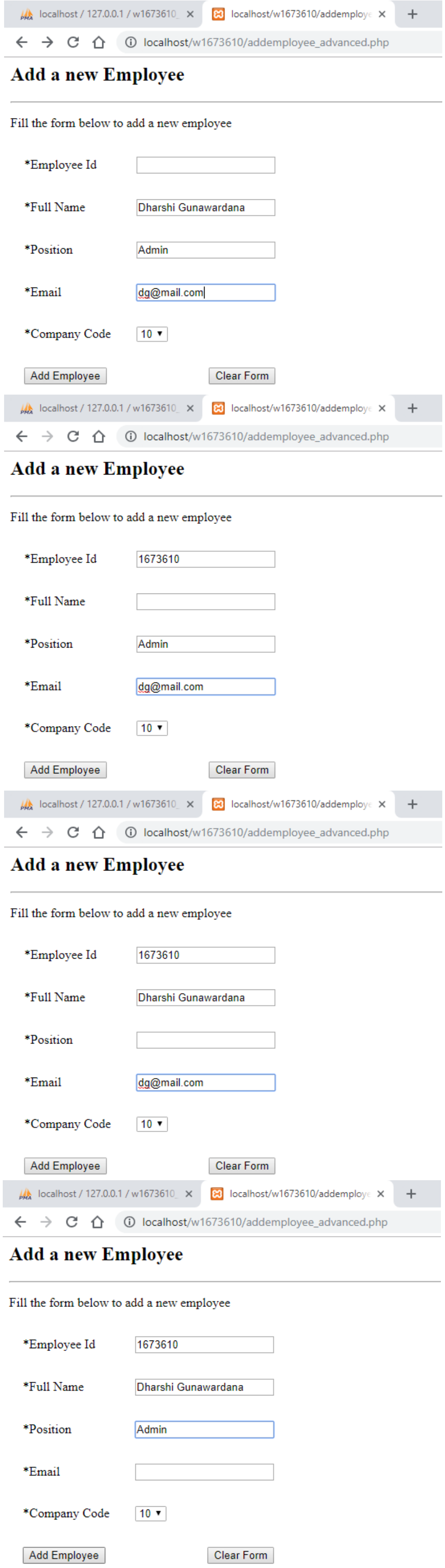
Print Copy to clipboard Export Display chart Create view

Last Record

Figure 18- Inserted data in Employee Table

3.5.1. Error Handling/ Validation

Figure 20- without filling field


localhost / 127.0.0.1 / w1673610
localhost/w1673610/addemploy
Add Employee
Clear Form

localhost/w1673610/addemployee_advanced.php

Add a new Employee

Fill the form below to add a new employee

*Employee Id	
*Full Name	Dharshi Gunawardana
*Position	Admin
*Email	dg@mail.com
*Company Code	10 ▾

localhost/w1673610/addemployee_advanced.php

Add a new Employee

Fill the form below to add a new employee

*Employee Id	1673610
*Full Name	
*Position	Admin
*Email	dg@mail.com
*Company Code	10 ▾

localhost/w1673610/addemployee_advanced.php

Add a new Employee

Fill the form below to add a new employee

*Employee Id	1673610
*Full Name	Dharshi Gunawardana
*Position	
*Email	dg@mail.com
*Company Code	10 ▾

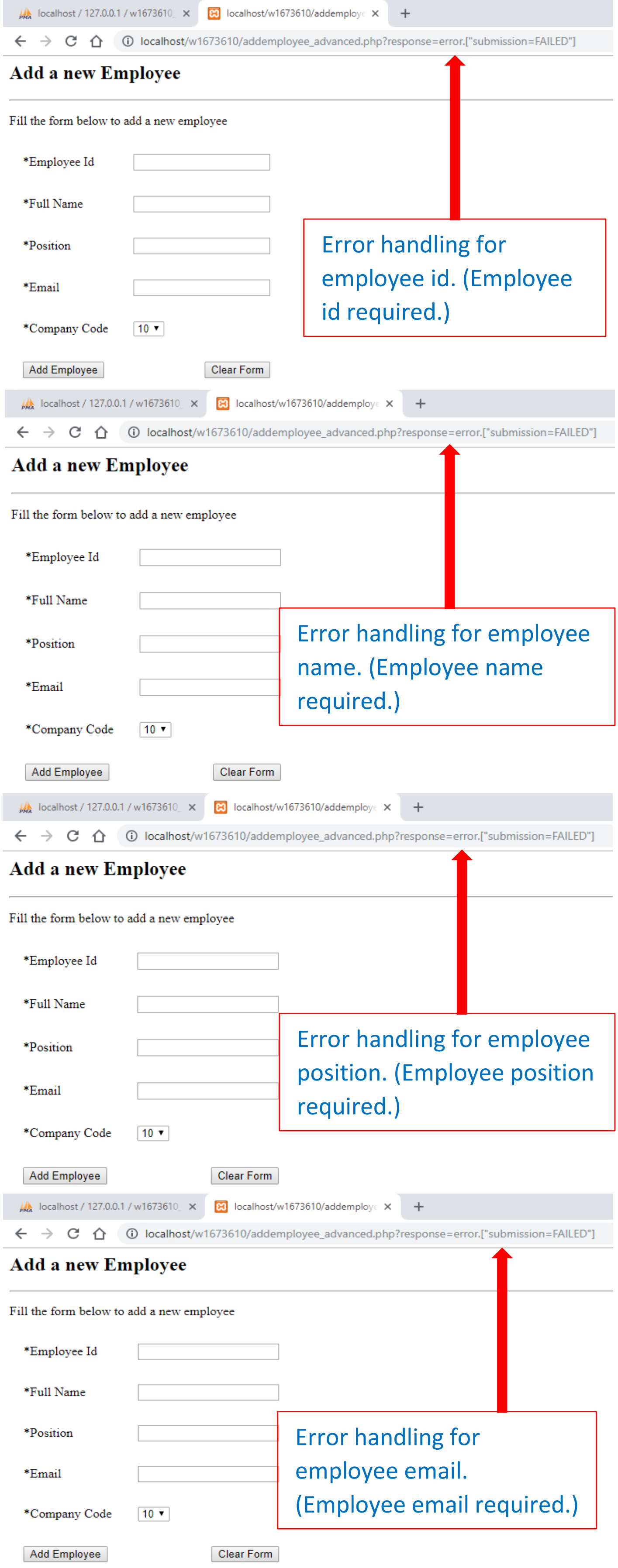
localhost/w1673610/addemployee_advanced.php

Add a new Employee

Fill the form below to add a new employee

*Employee Id	1673610
*Full Name	Dharshi Gunawardana
*Position	Admin
*Email	
*Company Code	10 ▾

Figure 19- show error message


localhost / 127.0.0.1 / w1673610
localhost/w1673610/addemploy
Add Employee
Clear Form

localhost/w1673610/addemployee_advanced.php?response=error.["submission=FAILED"]

Add a new Employee

Fill the form below to add a new employee

*Employee Id	
*Full Name	
*Position	
*Email	
*Company Code	10 ▾

Error handling for employee id. (Employee id required.)

localhost/w1673610/addemployee_advanced.php?response=error.["submission=FAILED"]

Add a new Employee

Fill the form below to add a new employee

*Employee Id	
*Full Name	
*Position	
*Email	
*Company Code	10 ▾

Error handling for employee name. (Employee name required.)

localhost/w1673610/addemployee_advanced.php?response=error.["submission=FAILED"]

Add a new Employee

Fill the form below to add a new employee

*Employee Id	
*Full Name	
*Position	
*Email	
*Company Code	10 ▾

Error handling for employee position. (Employee position required.)

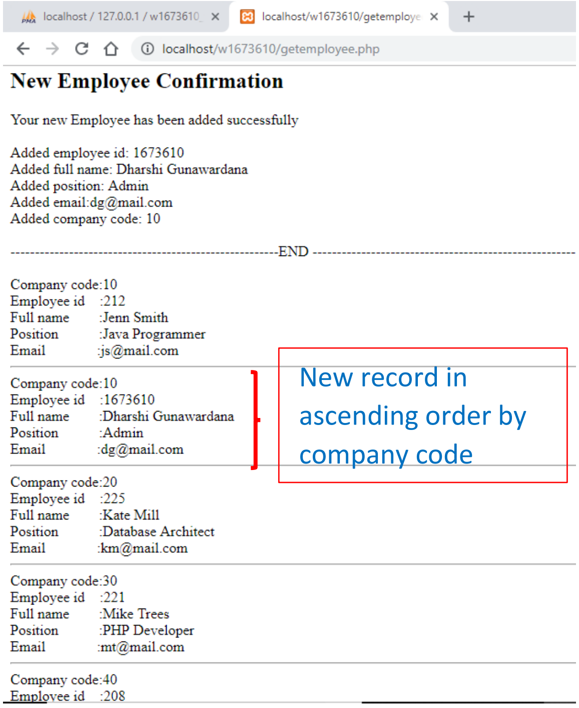
localhost/w1673610/addemployee_advanced.php?response=error.["submission=FAILED"]

Add a new Employee

Fill the form below to add a new employee

*Employee Id	
*Full Name	
*Position	
*Email	
*Company Code	10 ▾

Error handling for employee email. (Employee email required.)


localhost / 127.0.0.1 / w1673610_
localhost/w1673610/getemploye
localhost/w1673610/getemployee.php

New Employee Confirmation

Your new Employee has been added successfully

Added employee id: 1673610
Added full name: Dharshi Gunawardana
Added position: Admin
Added email:dg@mail.com
Added company code: 10

---END ---

Company code:10
Employee id :212
Full name :Jenn Smith
Position :Java Programmer
Email :js@mail.com

Company code:10
Employee id :1673610
Full name :Dharshi Gunawardana
Position :Admin
Email :dg@mail.com

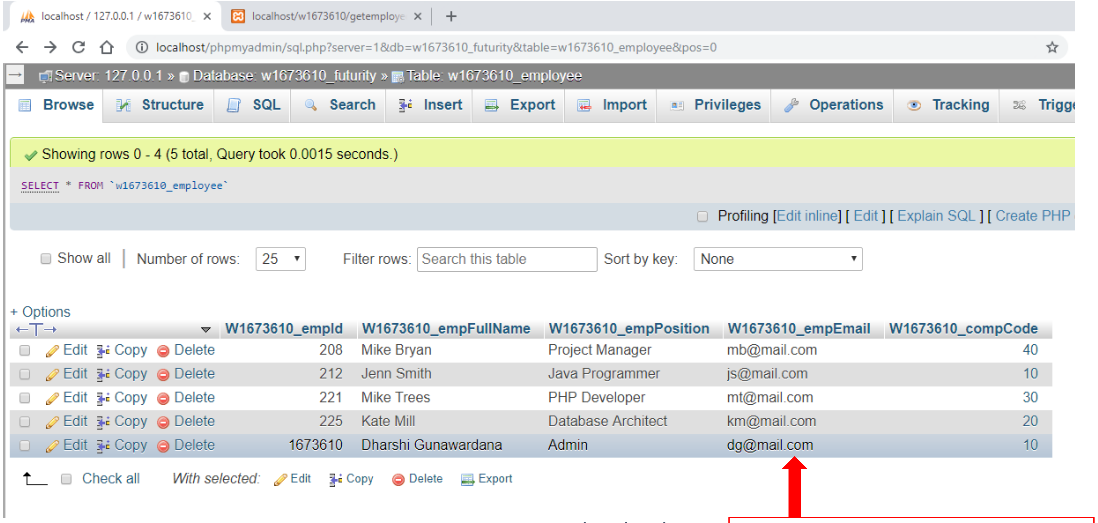
Company code:20
Employee id :225
Full name :Kate Mill
Position :Database Architect
Email :km@mail.com

Company code:30
Employee id :221
Full name :Mike Trees
Position :PHP Developer
Email :mt@mail.com

Company code:40
Employee id :208

New record in ascending order by company code

Figure 21- New record in getemployee.php


localhost / 127.0.0.1 / w1673610_
localhost/w1673610/getemploye
localhost/phpmyadmin/sql.php?server=1&db=w1673610_futurity&table=w1673610_employee&pos=0
Server: 127.0.0.1 » Database: w1673610_futurity » Table: w1673610_employee
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Trigge
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP
+ Options
Check all
With selected:
Edit
Copy
Delete
Export

✓ Showing rows 0 - 4 (5 total, Query took 0.0015 seconds.)

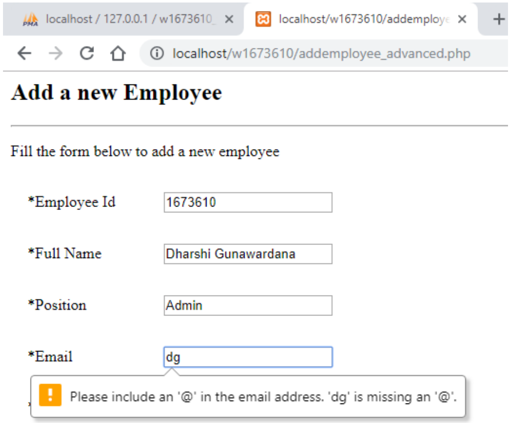
```
SELECT * FROM `w1673610_employee`
```

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

	W1673610_empId	W1673610_empFullName	W1673610_empPosition	W1673610_empEmail	W1673610_compCode
□ Edit Copy Delete	208	Mike Bryan	Project Manager	mb@mail.com	40
□ Edit Copy Delete	212	Jenn Smith	Java Programmer	js@mail.com	10
□ Edit Copy Delete	221	Mike Trees	PHP Developer	mt@mail.com	30
□ Edit Copy Delete	225	Kate Mill	Database Architect	km@mail.com	20
□ Edit Copy Delete	1673610	Dharshi Gunawardana	Admin	dg@mail.com	10

Figure 22- New Record in database

Successfully added New record


localhost / 127.0.0.1 / w1673610_
localhost/w1673610/addemploye
localhost/w1673610/addemployee_advanced.php
Add Employee
Clear Form

Add a new Employee

Fill the form below to add a new employee

*Employee Id 1673610

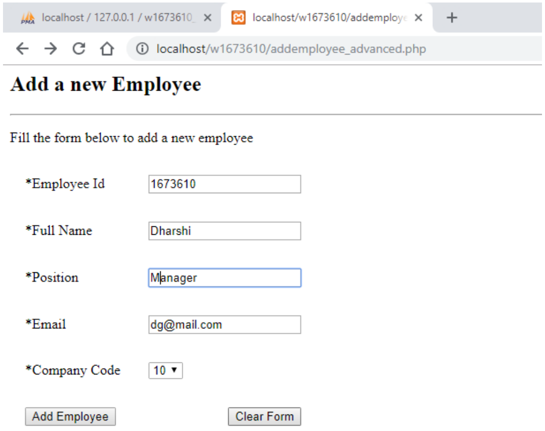
*Full Name Dharshi Gunawardana

*Position Admin

*Email dg

! Please include an '@' in the email address. 'dg' is missing an '@'.

Figure 23- Email Validation


localhost / 127.0.0.1 / w1673610_
localhost/w1673610/addemploye
localhost/w1673610/addemployee_advanced.php
Add Employee
Clear Form

Add a new Employee

Fill the form below to add a new employee

*Employee Id 1673610

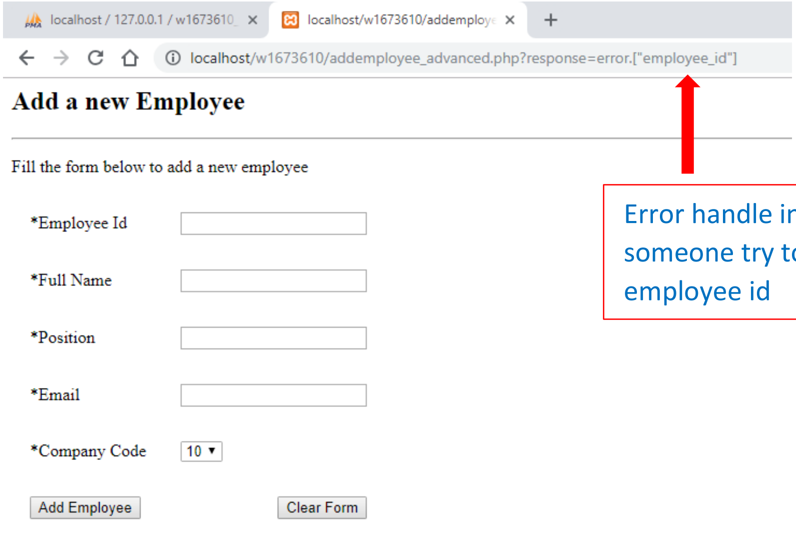
*Full Name Dharshi

*Position Manager

*Email dg@mail.com

*Company Code 10

Figure 24- Same emp_Id twice


localhost / 127.0.0.1 / w1673610
localhost/w1673610/addemploy
localhost/w1673610/addemployee_advanced.php?response=error.["employee_id"]

Add a new Employee

Fill the form below to add a new employee

*Employee Id

*Full Name

*Position

*Email

*Company Code 10 ▾

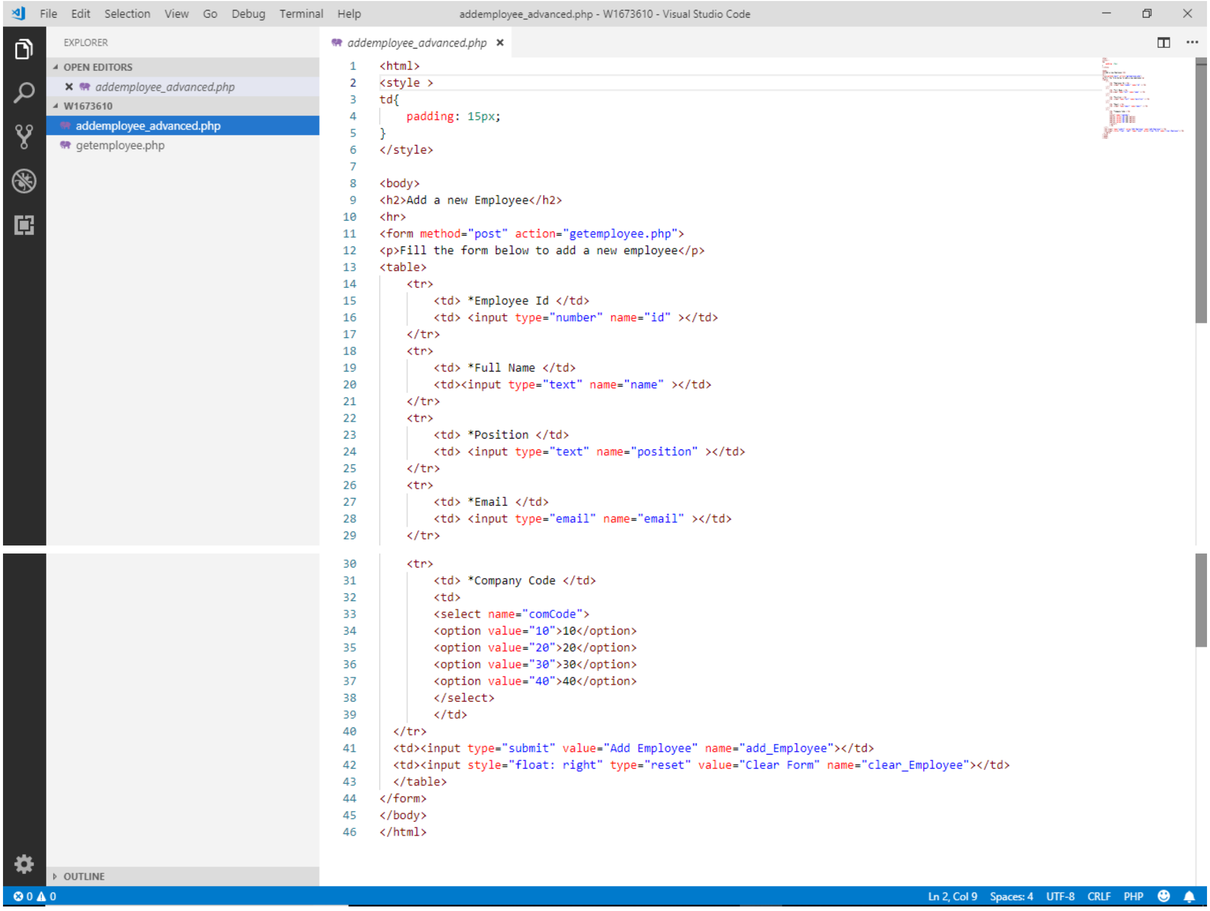
Add Employee   Clear Form

Figure 25- error handling

Error handle in header. If someone try to put same employee id

3.6. PHP code screenshot

3.6.1. addemployee_advanced.php


File Edit Selection View Go Debug Terminal Help
addemployee_advanced.php - W1673610 - Visual Studio Code
EXPLORER
OPEN EDITORS
addemployee_advanced.php
W1673610
addemployee_advanced.php
getemployee.php
OUTLINE
Ln 2, Col 9 Spaces: 4 UTF-8 CRLF PHP

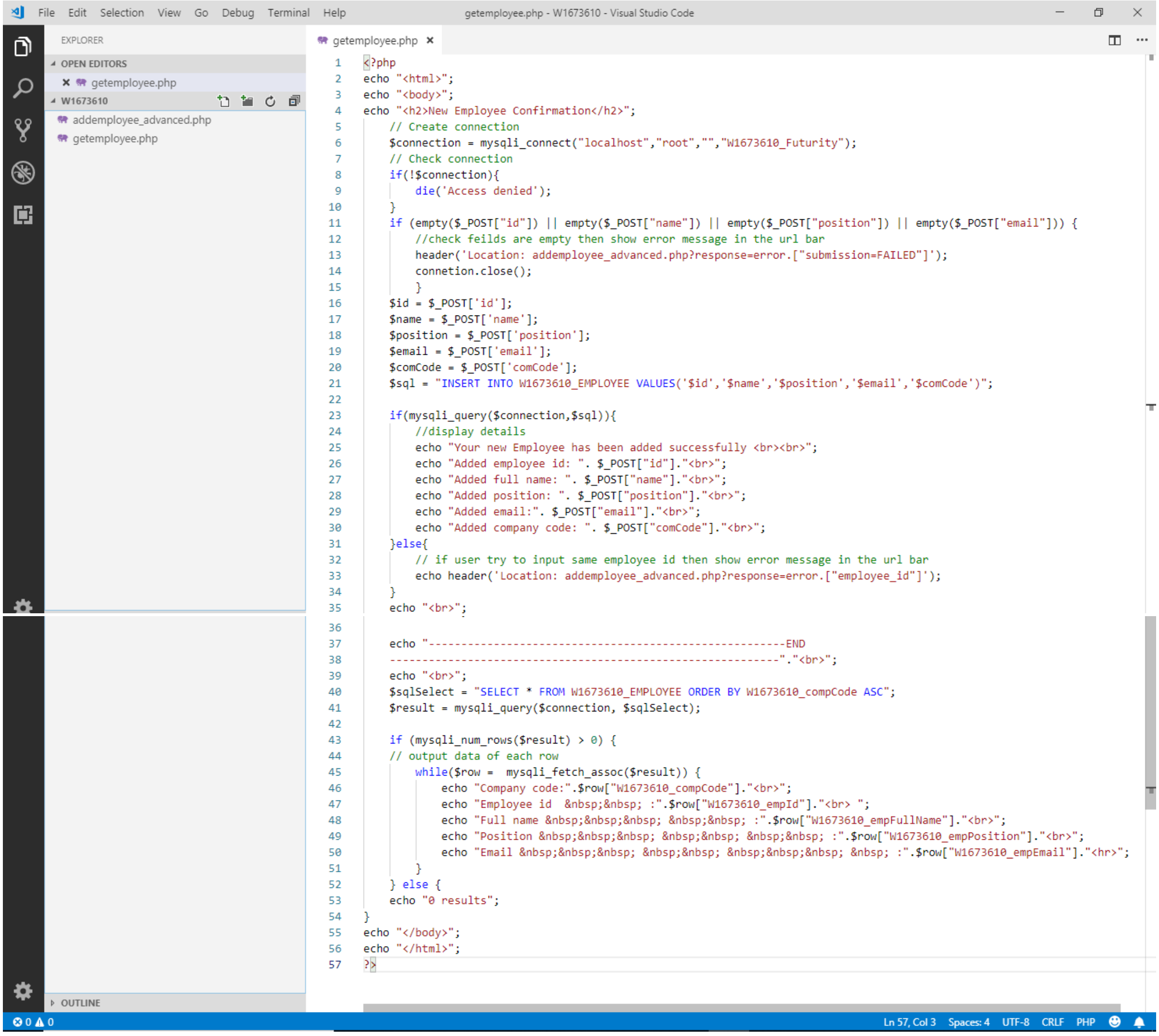
```
<html>
<style >
td{
    padding: 15px;
}
</style>

<body>
<h2>Add a new Employee</h2>
<hr>
<form method="post" action="getemployee.php">
<p>Fill the form below to add a new employee</p>
<table>
    <tr>
        <td> *Employee Id </td>
        <td> <input type="number" name="id" ></td>
    </tr>
    <tr>
        <td> *Full Name </td>
        <td><input type="text" name="name" ></td>
    </tr>
    <tr>
        <td> *Position </td>
        <td> <input type="text" name="position" ></td>
    </tr>
    <tr>
        <td> *Email </td>
        <td> <input type="email" name="email" ></td>
    </tr>

    <tr>
        <td> *Company Code </td>
        <td>
        <select name="comCode">
        <option value="10">10</option>
        <option value="20">20</option>
        <option value="30">30</option>
        <option value="40">40</option>
        </select>
        </td>
  </tr>
  <td><input type="submit" value="Add Employee" name="add_Employee"></td>
  <td><input style="float: right" type="reset" value="Clear Form" name="clear_Employee"></td>
  </table>
</form>
</body>
</html>
```

Figure 26- addemployee_advanced.php

3.6.2. getemployee.php



```php
<?php
echo "<html>";
echo "<body>";
echo "<h2>New Employee Confirmation</h2>";
    // Create connection
    $connection = mysqli_connect("localhost","root","","W1673610_Futurity");
    // Check connection
    if(!$connection){
        die('Access denied');
    }
    if (empty($_POST["id"]) || empty($_POST["name"]) || empty($_POST["position"]) || empty($_POST["email"])) {
        //check feilds are empty then show error message in the url bar
        header('Location: addemployee_advanced.php?response=error.["submission=FAILED"]');
        connetion.close();
        }
    $id = $_POST['id'];
    $name = $_POST['name'];
    $position = $_POST['position'];
    $email = $_POST['email'];
    $comCode = $_POST['comCode'];
    $sql = "INSERT INTO W1673610_EMPLOYEE VALUES('$id','$name','$position','$email','$comCode')";

    if(mysqli_query($connection,$sql)){
        //display details
        echo "Your new Employee has been added successfully <br><br>";
        echo "Added employee id: ". $_POST["id"]."<br>";
        echo "Added full name: ". $_POST["name"]."<br>";
        echo "Added position: ". $_POST["position"]."<br>";
        echo "Added email:". $_POST["email"]."<br>";
        echo "Added company code: ". $_POST["comCode"]."<br>";
    }else{
        // if user try to input same employee id then show error message in the url bar
        echo header('Location: addemployee_advanced.php?response=error.["employee_id"]');
    }
    echo "<br>";

    echo "-------------------------------------------------------END
    -------------------------------------------------------"."<br>";
    echo "<br>";
    $sqlSelect = "SELECT * FROM W1673610_EMPLOYEE ORDER BY W1673610_compCode ASC";
    $result = mysqli_query($connection, $sqlSelect);

    if (mysqli_num_rows($result) > 0) {
    // output data of each row
        while($row =  mysqli_fetch_assoc($result)) {
            echo "Company code:".$row["W1673610_compCode"]."<br>";
            echo "Employee id  &nbsp;&nbsp; :".$row["W1673610_empId"]."<br> ";
            echo "Full name &nbsp;&nbsp;&nbsp; &nbsp;&nbsp; :".$row["W1673610_empFullName"]."<br>";
            echo "Position &nbsp;&nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; :".$row["W1673610_empPosition"]."<br>";
            echo "Email &nbsp;&nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp;&nbsp; &nbsp; :".$row["W1673610_empEmail"]."<hr>";
        }
    } else {
    echo "0 results";
}
echo "</body>";
echo "</html>";
?>
```

Figure 27- getemployee.php

3.7. Appendices

3.7.1. Company, Employee tables in the MySQL RDBMS

```
/* Create Database*/

CREATE DATABASE  W1673610_Futurity;

USE  W1673610_Futurity;

/*Create Company  Table*/

CREATE TABLE W1673610_Company(

        W1673610_compCode           INT(2)          NOT NULL,

        W1673610_compName           VARCHAR(25)     NOT NULL,

        W1673610_compDescrip        VARCHAR(50)     NOT NULL,

  /*compCode is the primary key*/

        CONSTRAINT PK_Company PRIMARY KEY(W1673610_compCode),

  /*compName is Unique*/

  CONSTRAINT UK_compName UNIQUE(W1673610_compName)

);


/*Create Employee Table*/

CREATE TABLE W1673610_Employee(

        W1673610_empId                  INT             NOT NULL,

        W1673610_empFullName            VARCHAR(50)     NOT NULL,

        W1673610_empPosition            VARCHAR(25)     NOT NULL,

        W1673610_empEmail               VARCHAR(50)     NOT NULL,

        W1673610_compCode               INT(2)          NOT NULL,

  /*empEmail is Unique*/

  CONSTRAINT UK_empEmail UNIQUE(W1673610_empEmail),

  /*empId is the primary key*/

        CONSTRAINT PK_Employee PRIMARY KEY(W1673610_empId),

  /*compCode is the foreign key*/

  CONSTRAINT FK_Company FOREIGN KEY(W1673610_compCode) REFERENCES
W1673610_Company(W1673610_compCode)

);
```

```
/*Insert 4 records to Company Table*/

INSERT INTO W1673610_Company VALUES(10,"IIT","Informatics Institute of Technology");

INSERT INTO W1673610_Company VALUES(20,"AIT","Asian Institute of Technology");

INSERT INTO W1673610_Company VALUES(30,"BIT","Bachelor of Information Technology");

INSERT INTO W1673610_Company VALUES(40,"IT","Informatic Technology");

/* Select data from Company */

SELECT * FROM w1673610_Company;

/* Select data from Employee */

SELECT * FROM w1673610_Employee;
```

3.7.2. Code for addemployee_advanced.php

```
<html>

<style >
td
{
    padding: 15px;
}
</style>

<body>

<h2>Add a new Employee</h2>
<hr>
<form method="post" action="getemployee.php">
<p>Fill the form below to add a new employee</p>

<table>
    <tr>
        <td> *Employee Id </td>
        <td> <input type="number" name="id" ></td>
    </tr>
    <tr>
        <td> *Full Name </td>
        <td><input type="text" name="name" ></td>
    </tr>
    <tr>
        <td> *Position </td>
        <td> <input type="text" name="position" ></td>
    </tr>
    <tr>
        <td> *Email </td>
        <td> <input type="email" name="email" ></td>
    </tr>
    <tr>
        <td> *Company Code </td>
        <td>
        <select name="comCode">
        <option value="10">10</option>
        <option value="20">20</option>
        <option value="30">30</option>
        <option value="40">40</option>
        </select>
        </td>
  </tr>
  <td><input type="submit" value="Add Employee" name="add_Employee"></td>
  <td><input style="float: right" type="reset" value="Clear Form"
name="clear_Employee"></td>
  </table>
</form>
</body>
</html>
```

3.7.3. Code for getemployee.php

```
<?php

echo "<html>";
echo "<body>";
echo "<h2>New Employee Confirmation</h2>";
    // Create connection
    $connection = mysqli_connect("localhost","root","","W1673610_Futurity");
    // Check connection
    if(!$connection){
        die('Access denied');
    }

    if (empty($_POST["id"]) || empty($_POST["name"]) ||
empty($_POST["position"]) || empty($_POST["email"])) {
        //check fields are empty then show error message in the url bar
        header('Location: addemployee_advanced.php?response=error.
["submission=FAILED"]');
        connetion.close();
        }

    $id = $_POST['id'];
    $name = $_POST['name'];
    $position = $_POST['position'];
    $email = $_POST['email'];
    $comCode = $_POST['comCode'];

    $sql = "INSERT INTO W1673610_EMPLOYEE VALUES('$id','$name','$position',
'$email','$comCode')";
```

```
    if(mysqli_query($connection,$sql)){
        //display details
        echo "Your new Employee has been added successfully <br><br>";
        echo "Added employee id: ". $_POST["id"]."<br>";
        echo "Added full name: ". $_POST["name"]."<br>";
        echo "Added position: ". $_POST["position"]."<br>";
        echo "Added email:". $_POST["email"]."<br>";
        echo "Added company code: ". $_POST["comCode"]."<br>";
    }else{
// if user try to input same employee id then show error message in the url
bar
        echo header('Location: addemployee_advanced.php?response=error.
["employee_id"]');
    }
    echo "<br>";

    echo "-------------------------------------------------------END
    ------------------------------------------------------------"."<br>";
    echo "<br>";
    $sqlSelect = "SELECT * FROM W1673610_EMPLOYEE ORDER BY W1673610_compCode
ASC";
    $result = mysqli_query($connection, $sqlSelect);

    if (mysqli_num_rows($result) > 0) {
    // output data of each row
        while($row =  mysqli_fetch_assoc($result)) {
            echo "Company code:".$row["W1673610_compCode"]."<br>";

            echo "Employee id  &nbsp;&nbsp; :".$row["W1673610_empId"]."<br> ";

            echo "Full name &nbsp;&nbsp;&nbsp; &nbsp;&nbsp;
:".$row["W1673610_empFullName"]."<br>";

            echo "Position &nbsp;&nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp;
:".$row["W1673610_empPosition"]."<br>";

            echo "Email &nbsp;&nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp;&nbsp;
&nbsp; :".$row["W1673610_empEmail"]."<hr>";

        }
    } else {
            echo "0 results";
      }
echo "</body>";
echo "</html>";
?>
```