Course	:	BIC 21404 Database
Session	:	II 2024/2025
Lab sheet	:	2
Objective	:	At the end of the session, students are able to: i. Create table using command line ii. Alter tables iii. Inserting data into tables
Lab activity	:	• Create table using command line • Alter tables • Inserting data into tables

Introduction to Schema Objects

A database schema is a logical container for data structures, called schema objects. Examples of schema objects are tables and indexes. Schema objects are created and manipulated with SQL.

A database user has a password and various database privileges. Each user owns a single schema, which has the same name as the user. The schema contains the data for the user owning the schema. For example, the hr user owns the hr schema, which contains schema objects such as the employees table. In a production database, the schema owner usually represents a database application rather than a person.

Within a schema, each schema object of a particular type has a unique name. For example, hr.employees refers to the table employees in the hr schema. Figure 2-1 depicts a schema owner named hr and schema objects within the hr schema.

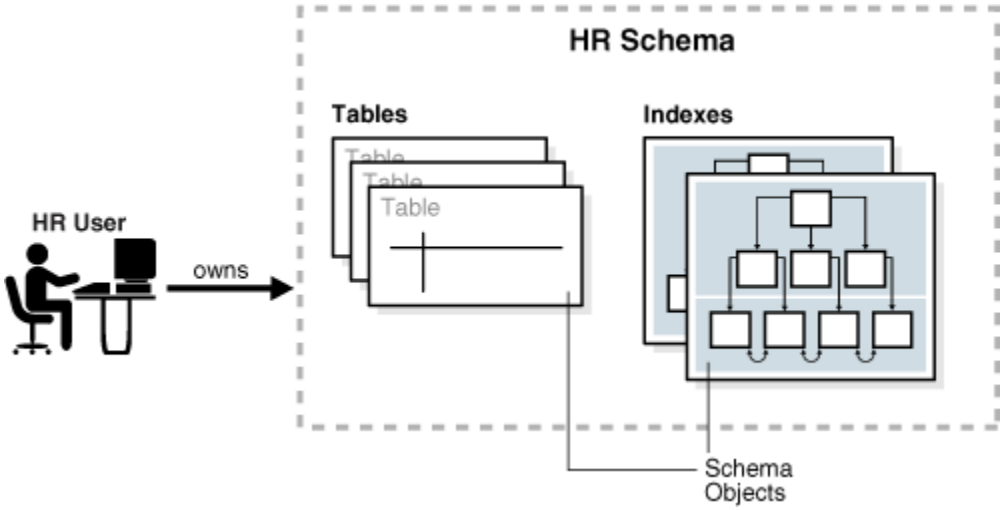


Figure 2-1: HR Schema

The hr schema is a sample schema that contains information about employees, departments and locations, work histories, and so on. The following Figure 2-2 is an entity-relationship diagram of the tables in the hr schema.

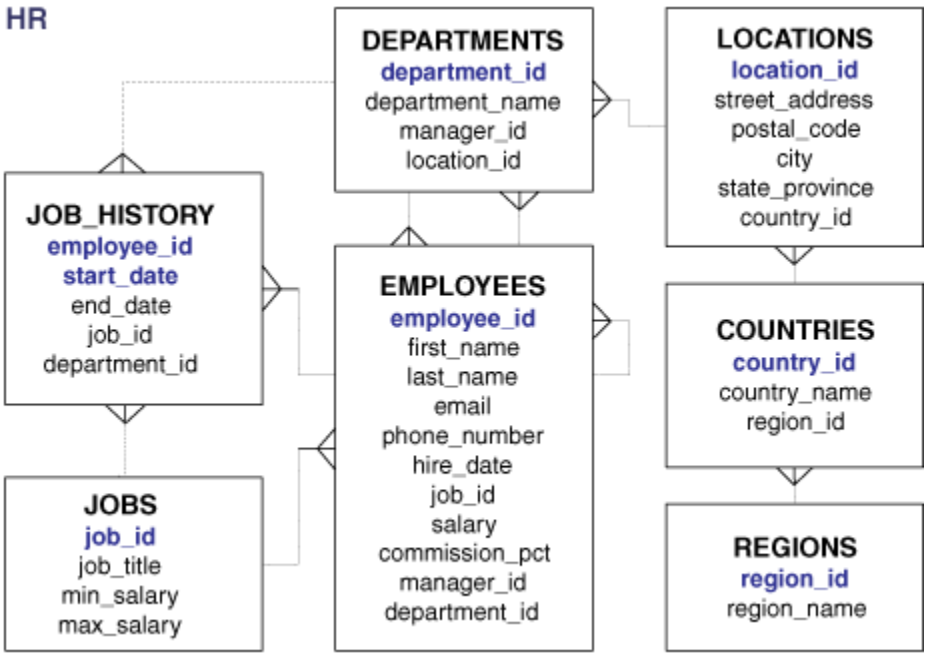


Figure 2-2: HUMAN RESOURCE (hr) Schema

Overview of Tables

A table is the basic unit of data organization in an Oracle database. A table describes an entity, which is something of significance about which information must be recorded. For example, an employee could be an entity.

A table definition includes a table name and set of columns. A column identifies an attribute of the entity described by the table. For example, the column employee_idin the employees table refers to the employee ID attribute of an employee entity.

In general, you give each column a column name, a data type, and a width when you create a table. For example, the data type for employee_id is INT (6), indicating that this column can only contain numeric data up to 6 digits in width. The width can be predetermined by the data type, as in the case of DATE.

A table can contain a virtual column, which unlike a nonvirtual column does not consume disk space. The database derives the values in a virtual column on demand by computing a set of user-specified expressions or functions. For example, the virtual column income could be a function of the salary and commission_pct columns.

After you create a table, you can insert, query, delete, and update rows using SQL. A row is a collection of column information corresponding to a record in a table. For example, a row in the employees table describes the attributes of a specific employee.

Table 1: HUMANRESOURCE, hr Table Descriptions

Table DEPARTMENTS

Name	Null?	Type
DEPARTMENT_ID (*primary key)	NOT NULL	INT(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR(30)
MANAGER_ID		INT(6)
LOCATION_ID		INT(4)

Table EMPLOYEES

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	INT(6)
FIRST_NAME		VARCHAR(20)
LAST_NAME	NOT NULL	VARCHAR(25)
EMAIL	NOT NULL	VARCHAR(25)
PHONE_NUMBER		VARCHAR(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR(10)
SALARY		INT(8,2)
COMMISSION_PCT		INT(2,2)
MANAGER_ID		INT(6)
DEPARTMENT_ID		INT(4)

Table JOBS

Name	Null?	Type
JOB_ID	NOT NULL	VARCHAR(10)
JOB_TITLE	NOT NULL	VARCHAR(35)
MIN_SALARY		INT(6)
MAX_SALARY		INT(6)

Table JOB_HISTORY

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	INT(6)
START_DATE	NOT NULL	DATE
END_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR(10)
DEPARTMENT_ID		INT(4)

Table LOCATIONS

Name	Null?	Type
LOCATION_ID	NOT NULL	INT(4)
STREET_ADDRESS		VARCHAR(40)
POSTAL_CODE		VARCHAR(12)
CITY	NOT NULL	VARCHAR(30)
STATE_PROVINCE		VARCHAR(25)
COUNTRY_ID		CHAR(2)

Table REGIONS

Name	Null?	Type
REGION_ID	NOT NULL	INT
REGION_NAME		VARCHAR(25)

Table COUNTRIES

Name	Null?	Type
COUNTRY_ID	NOT NULL	CHAR(2)
COUNTRY_NAME		VARCHAR(40)
REGION_ID		INT

Unit 1: Creating a Database

i. Activate XAMPP and phpMyAdmin.
ii. Click on the **SQL** tab in PhpMyAdmin.
iii. In the empty box, type the following statement

```
CREATE DATABASE IF NOT EXISTS hr;
```

iv. Click **Go**
v. Check the new created database, hr.

Unit 2: Creating table(s)

i. Click on the SQL tab. In the empty box type the following

```
USE hr;

CREATE TABLE DEPARTMENTS (
DEPARTMENT_ID INT(4) NOT NULL AUTO_INCREMENT,
DEPARTMENT_NAME VARCHAR(30) NOT NULL,
MANAGER_ID INT(6),
```

```
LOCATION_ID INT(4),
PRIMARY KEY (DEPARTMENT_ID)
);
```

ii. Fix errors if any.
iii. Refresh the page and check newly created table in hr database.
iv. Using the same step in (i), create all tables listed in Table 1.

DEPARTMENT SQL

```
DESC departments;
```

[Edit inline] [Edit] [Create PHP code]

Extra options

Field	Type	Null	Key	Default	Extra
DEPARTMENT_ID	int(4)	NO	PRI	*NULL*	auto_increment
DEPARTMENT_NAME	varchar(30)	NO		*NULL*	
MANAGER_ID	int(6)	YES		*NULL*	
LOCATION_ID	int(4)	YES		*NULL*	

EMPLOYEES SQL

Your SQL query has been executed successfully.

```
DESC EMPLOYEES;
```

[Edit inline] [Edit] [Create PHP code]

Extra options

Field	Type	Null	Key	Default	Extra
EMPLOYEE_ID	int(6)	NO		*NULL*	
FIRST_NAME	varchar(20)	YES		*NULL*	
LAST_NAME	varchar(25)	NO		*NULL*	
EMAIL	varchar(20)	NO		*NULL*	
PHONE_NUMBER	varchar(20)	YES		*NULL*	
HIRE_DATE	date	NO		*NULL*	
JOB_ID	varchar(10)	NO		*NULL*	
SALARY	decimal(10,2)	YES		*NULL*	
COMMISSION_PCT	decimal(4,2)	YES		*NULL*	
MANAGER_ID	int(6)	YES		*NULL*	
DEPARTMENT_ID	int(4)	YES		*NULL*	

JOBS SQL

```
DESC JOBS;
```

[Edit inline] [Edit] [Create PHP code]

Extra options

Field	Type	Null	Key	Default	Extra
JOB_ID	varchar(10)	NO		*NULL*	
JOB_TITLE	varchar(35)	NO		*NULL*	
MIN_SALARY	int(6)	YES		*NULL*	
DEPARTMENT_ID	int(6)	YES		*NULL*	

JOBS HISTORY SQL

```
DESC job_history;
```

[Edit inline] [Edit] [Create PHP code]

Extra options

Field	Type	Null	Key	Default	Extra
EMPLOYEE_ID	int(6)	NO		*NULL*	
START_DATE	date	NO		*NULL*	
END_DATE	date	NO		*NULL*	
JOB_ID	varchar(10)	NO		*NULL*	
DEPARTMENT_ID	int(4)	YES		*NULL*	

LOCATION SQL

```
DESC LOCATIONS;
```

[Edit inline] [Edit] [Create PHP code]

Extra options

Field	Type	Null	Key	Default	Extra
LOCATION_ID	int(4)	NO		*NULL*	
STREET_ADDRESS	varchar(40)	YES		*NULL*	
POSTAL_CODE	varchar(12)	YES		*NULL*	
CITY	varchar(30)	NO		*NULL*	
STATE_PROVINCE	varchar(25)	YES		*NULL*	
COUNTRY_ID	char(2)	YES		*NULL*	

REGIONS SQL

```
DESC REGIONS;
```

[Edit inline] [Edit] [Create PHP code]

Extra options

Field	Type	Null	Key	Default	Extra
REGION_ID	int(11)	NO		*NULL*	
REGION_NAME	varchar(25)	YES		*NULL*	

COUNTRIES SQL

```
DESC countries;
```

[Edit inline] [Edit] [Create PHP code]

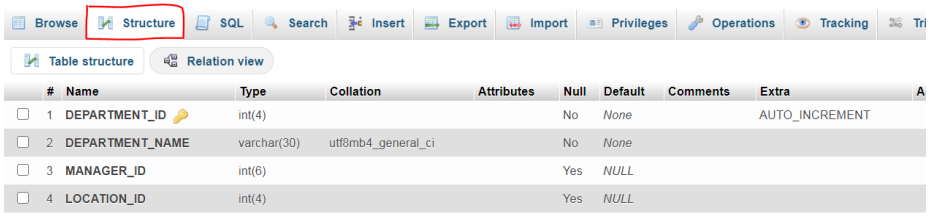
Extra options

Field	Type	Null	Key	Default	Extra
COUNTRY_ID	char(2)	NO		*NULL*	
COUNTRY_NAME	varchar(40)	YES		*NULL*	
REGION_ID	int(11)	YES		*NULL*	

Unit 3: Displaying Table Structure

Check table structure for all created tables for hr database

1. Click on the table, and select *structure* tab



Note
• DEFAULT is used to give a default value for the field when entering a new record. • NULL means that the field can be empty. NOT NULL means the opposite. • PRIMARY KEY is used to specify the field name which is to be used as a primary key. • KEY is normally a synonym for INDEX. This is usually used to identify fields in a table which can be linked to primary keys in other tables. • ENGINE=InnoDB: specifies the MySQL database engine as there are several MySQL engines.

- AUTO_INCREMENT: specifies which field is an auto-generated number.
- AUTO_INCREMENT=1: the first number to start with.
- CHARSET is a synonym for CHARACTER SET. MySQL allows storing data using a variety of
  character sets and to perform comparisons according to a variety of collations. For more information search for "MySQL charset".
- UNIQUE: creates a constraint such that all values in the field must be distinct. However, in most MySQL engines, unique fields can be null. This makes it different from a primary key.

Data types

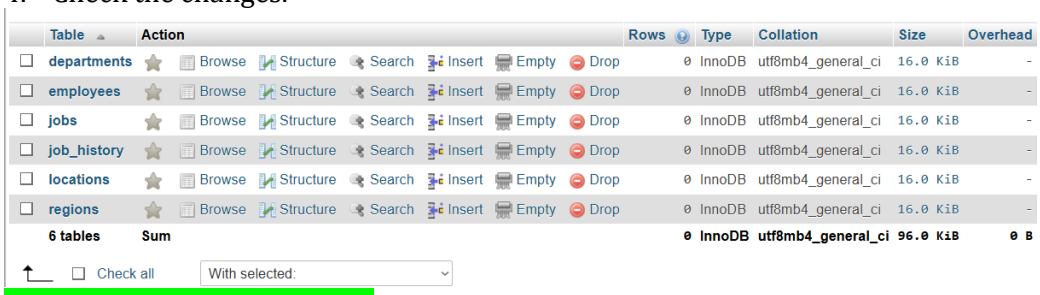
```
INT[(length)] [UNSIGNED] [ZEROFILL]
INTEGER[(length)] [UNSIGNED] [ZEROFILL]
BIGINT[(length)] [UNSIGNED] [ZEROFILL]
REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]
NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]
DATE
TIME
TIMESTAMP
DATETIME
YEAR
CHAR[(length)]
[CHARACTER SET charset_name] [COLLATE collation_name]
VARCHAR(length)
[CHARACTER SET charset_name] [COLLATE collation_name]
BINARY[(length)]
VARBINARY(length)
TINYBLOB
BLOB
MEDIUMBLOB
LONGBLOB
TINYTEXT [BINARY]
[CHARACTER SET charset_name] [COLLATE collation_name]
TEXT [BINARY]
[CHARACTER SET charset_name] [COLLATE collation_name]
MEDIUMTEXT [BINARY]
[CHARACTER SET charset_name] [COLLATE collation_name]
LONGTEXT [BINARY]
[CHARACTER SET charset_name] [COLLATE collation_name]
ENUM(value1,value2,value3,...)
[CHARACTER SET charset_name] [COLLATE collation_name]
SET(value1,value2,value3,...)
[CHARACTER SET charset_name] [COLLATE collation_name]
```

	Table	Action	Rows	Type	Collation	Size	Overhead
☐	countries	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	departments	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	employees	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	job_history	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	locations	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	regions	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	7 tables	Sum	0	InnoDB	utf8mb4_general_ci	112.0 KiB	0 B

Unit 4: Delete Table

1. On the hr database, select SQL tab.
2. Write the following statement:
   ```
   DROP TABLE table_name;
   ```

3. Click **Go**
4. Check the changes.



I DELETE TABLE COUNTRIES

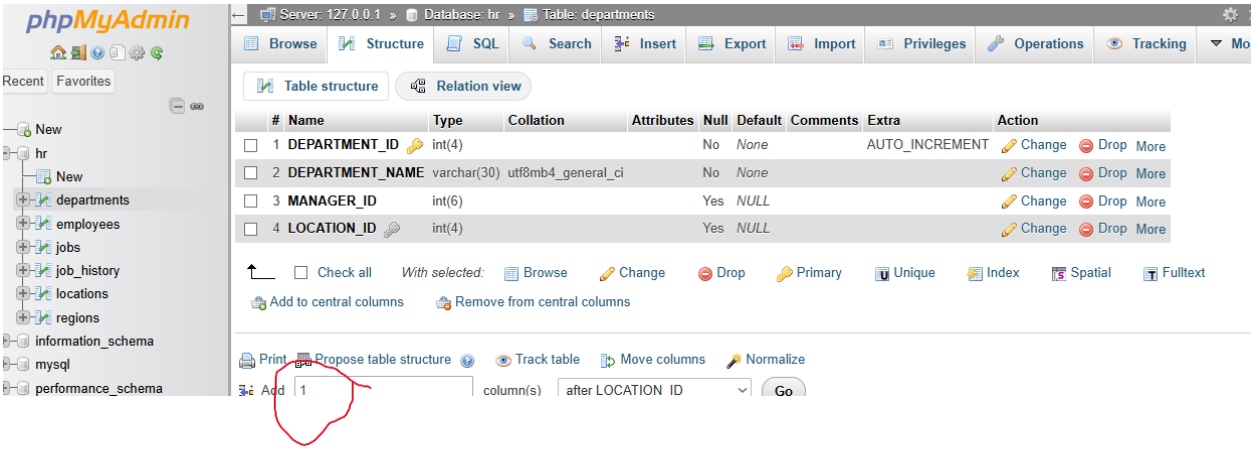
Unit 5: Update Tables

1. Use ALTER function to update table information/structure/constraints etc.
2. Example: Create Foreign Key constraints
   This exercise allows you to create a constraint to a table by creating a primary – foreign key constraint.

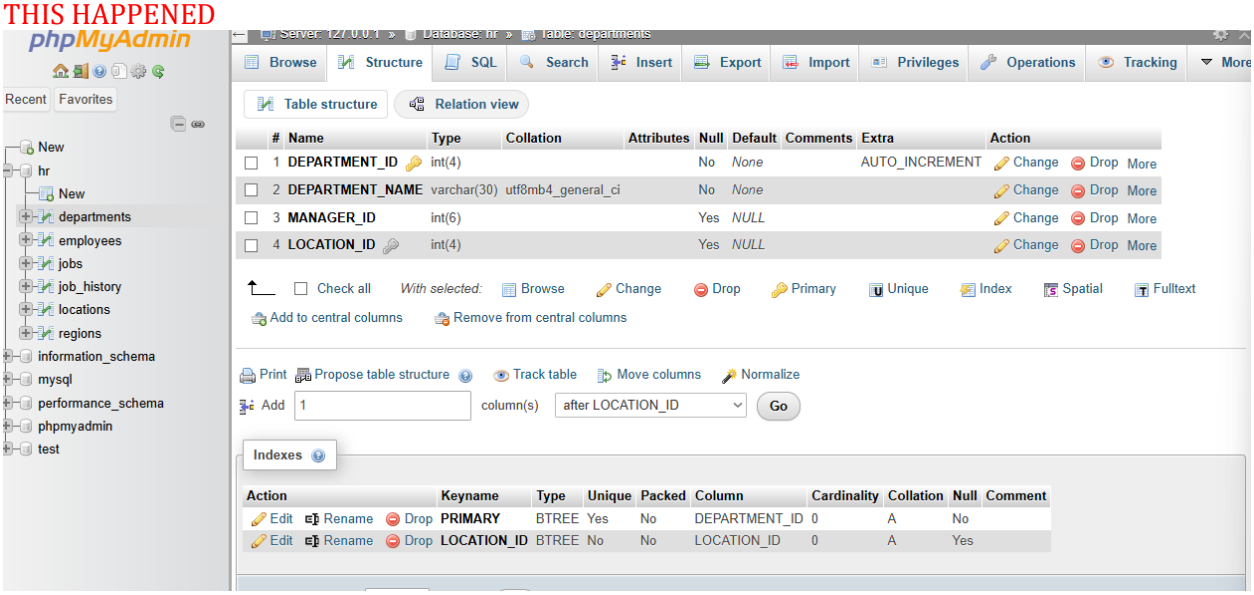
3. To link the departments table and location table, we use foreign key concept.
4. Run SQL query/queries on database hr:
5. Write the following statement

   ```
   ALTER TABLE departments
   ADD CONSTRAINT FOREIGN KEY (LOCATION_ID)
   REFERENCES departments (department_id) ON DELETE CASCADE on update cascade
   ```

6. Check the changes.



THIS HAPPENED

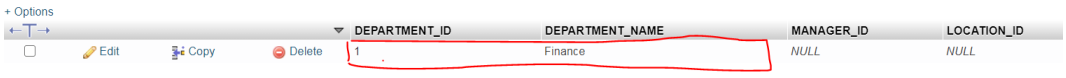


Unit 6: Insert Data Into Tables

1. Use INSERT INTO command to insert data into tables.
2. Example; one record is inserted into *departments* table.
3. On the hr database, select SQL tab.
4. Write the following statement

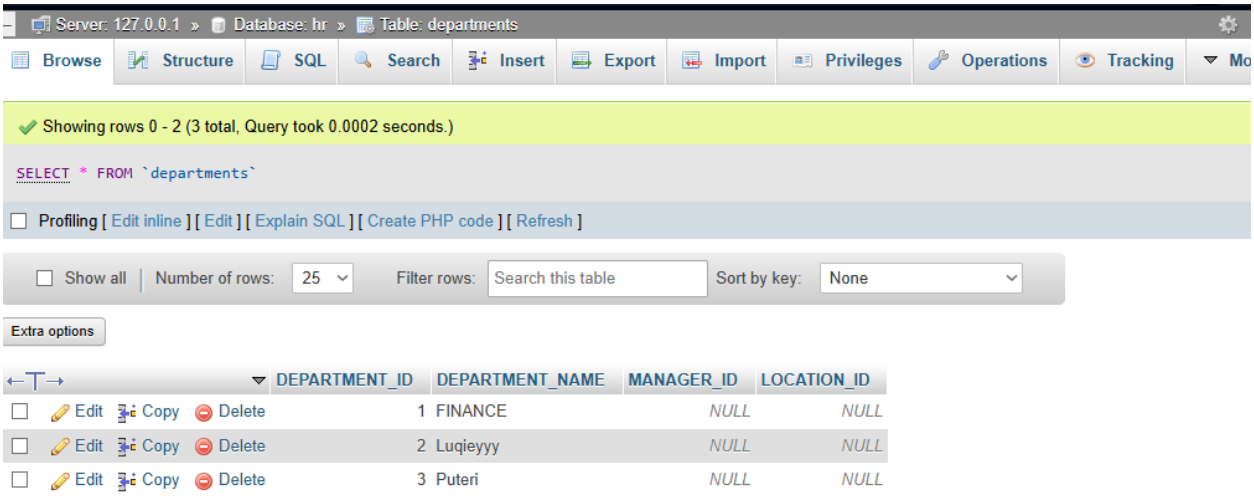
```
INSERT INTO departments (DEPARTMENT_ID, DEPARTMENT_NAME)
VALUES (001, "Finance")
```

5. Check the changes. New data record is inserted in the *departments* table.

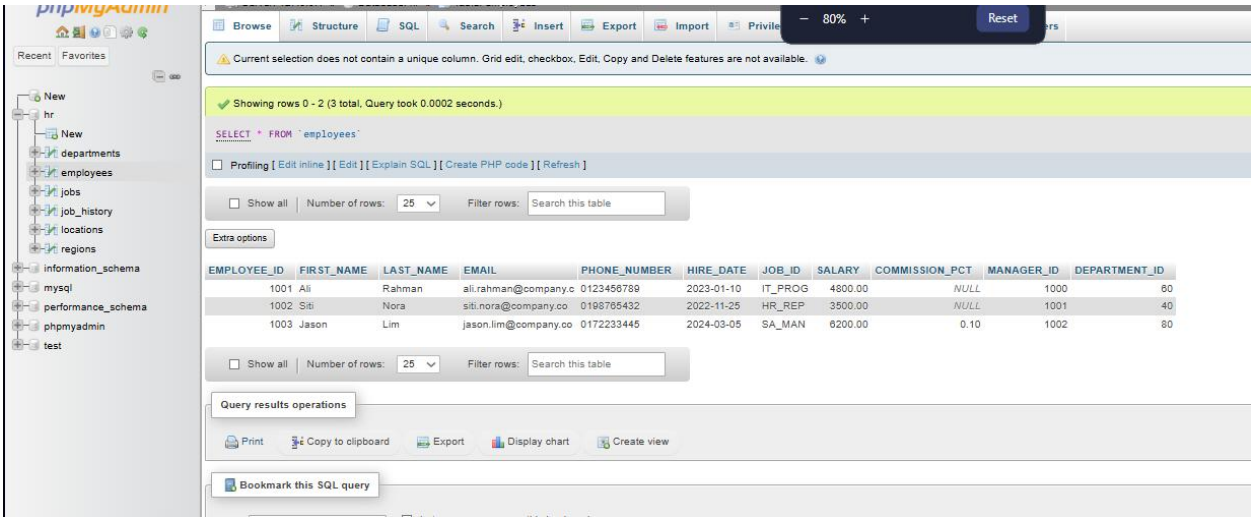


6. Add 3 data in ALL the tables created in Unit 2.

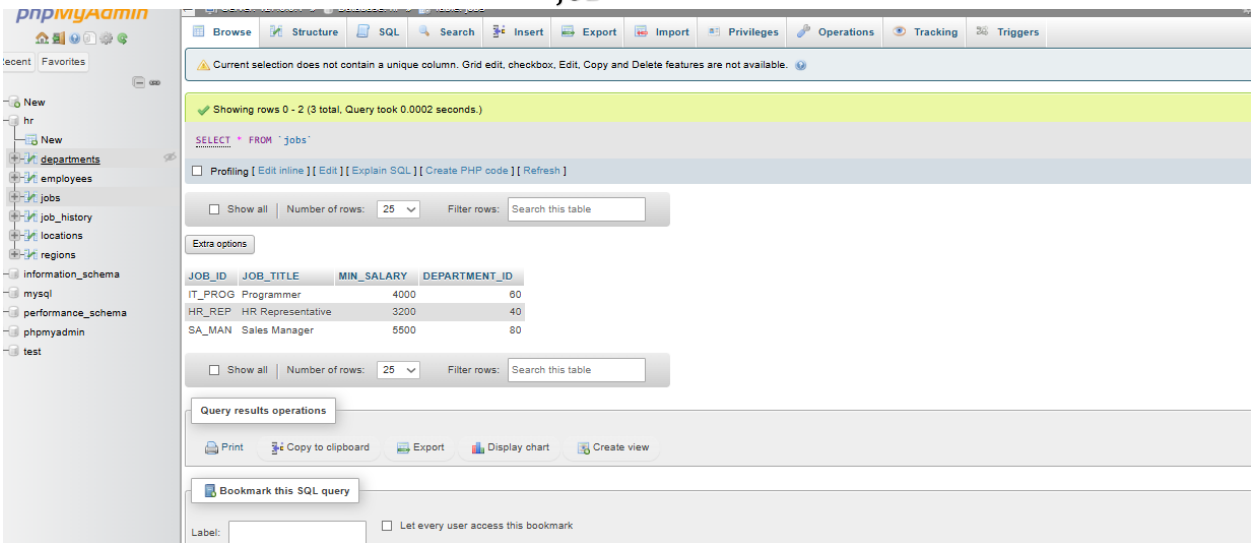
DEPARTMENT



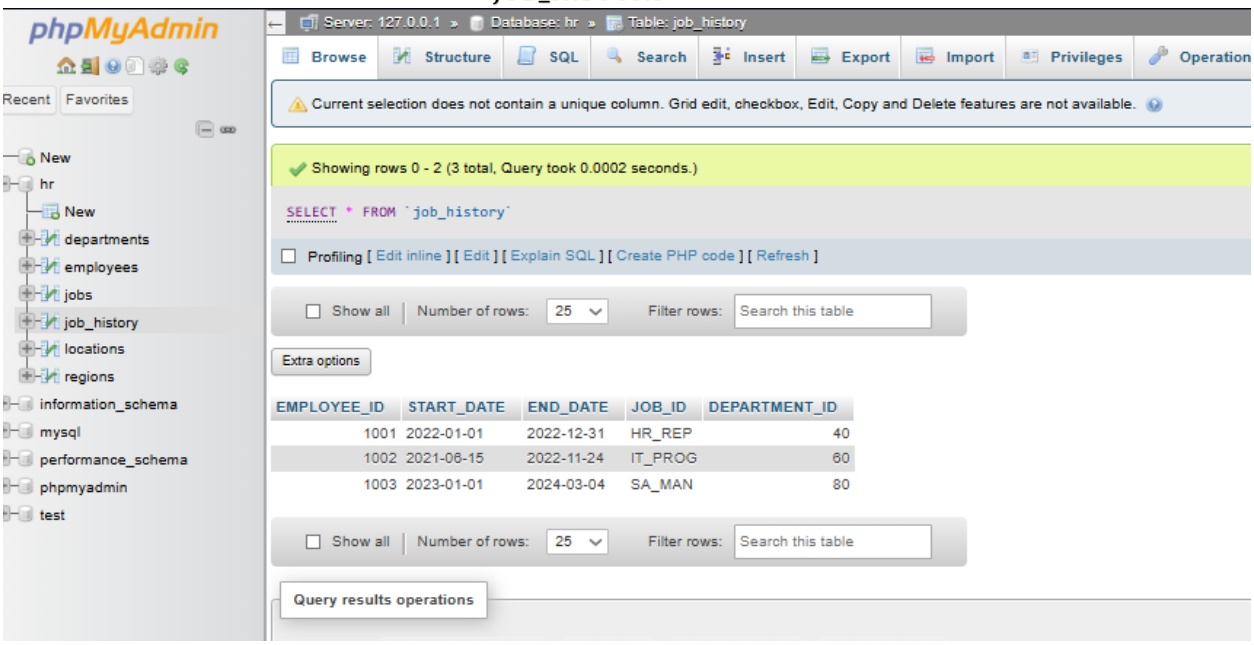
EMPLOYEES



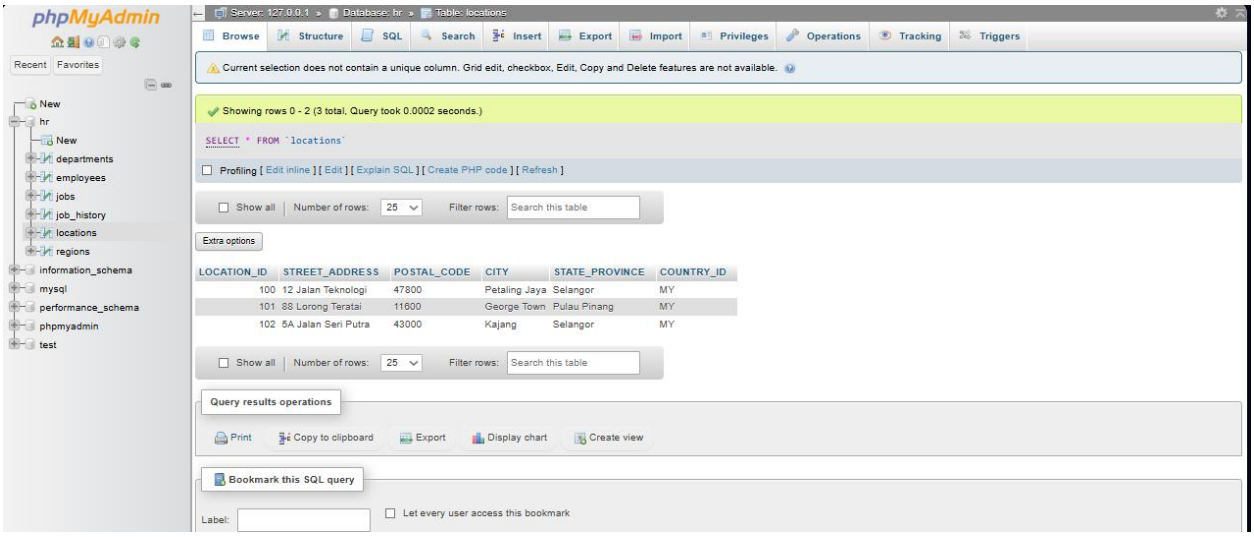
JOB



JOB_HISTORY



LOCATIONS



REGIONS