TOPIC:

A DESIGN AND IMPLEMENTATION OF A RAILWAY INFORMATION SYSTEM

Submitted to

Basic Sciences and Humanities Department
Of
Cyprus International University
Faculty of Engineering
Department Of Computer And Programming Engineering

DATABASE MANAGEMENT SYSTEMS AND PROGRAMMING I (CMP343)

By

DJOUKAN TCHINDA WILLY STEVEN
22016432
&
JOSEPHINE YANKA-BABU JACOBS
22002653

Saturday 14th January, 2023.

TABLE OF CONTENT

1- INTRODUCTION

The main aim of this report is to develop a system that provide facility for reporting relevant information about a railway system. Actually, through this system, we can retrieve data about the railway company like the number of passengers that did a reservation or bought a ticket on a specific date, the routes each train take on daily basis. We can also find the busy schedules, rush hours, most preferred routes the trains take daily, one of the station representative can track the details of the coaches working hours, his availability on a given date and time.

Furthermore, each table is related to one another which is represented on the entity relational diagram (ERD) shown on figure 1 and through the relational model we can easily see their associativity show on figure 2.

The database was created by the Data Defintion Language (DDL) which was named "**railway_info_sys**" and we recorded the data through the Data Manipulation Language (DML) for each table respectively.

The normalization of the tables on the other hand, was been done through the 1NF, 2NF and 3NF. This is to find out any anomaly in the database to prevent mostly redundancy and waste of space. By using this technique, it organizes data into groups to form stable, flexible and adaptive entities for the company which enhances their efficiency.

This system is develop to ease the management of the railway company, to be able to trace or retrieve every details when need in the near future either for update, modify, alter, insert or delete any detail in the database.

2- ENTITY RELATIONAL DIAGRAM (ERD)

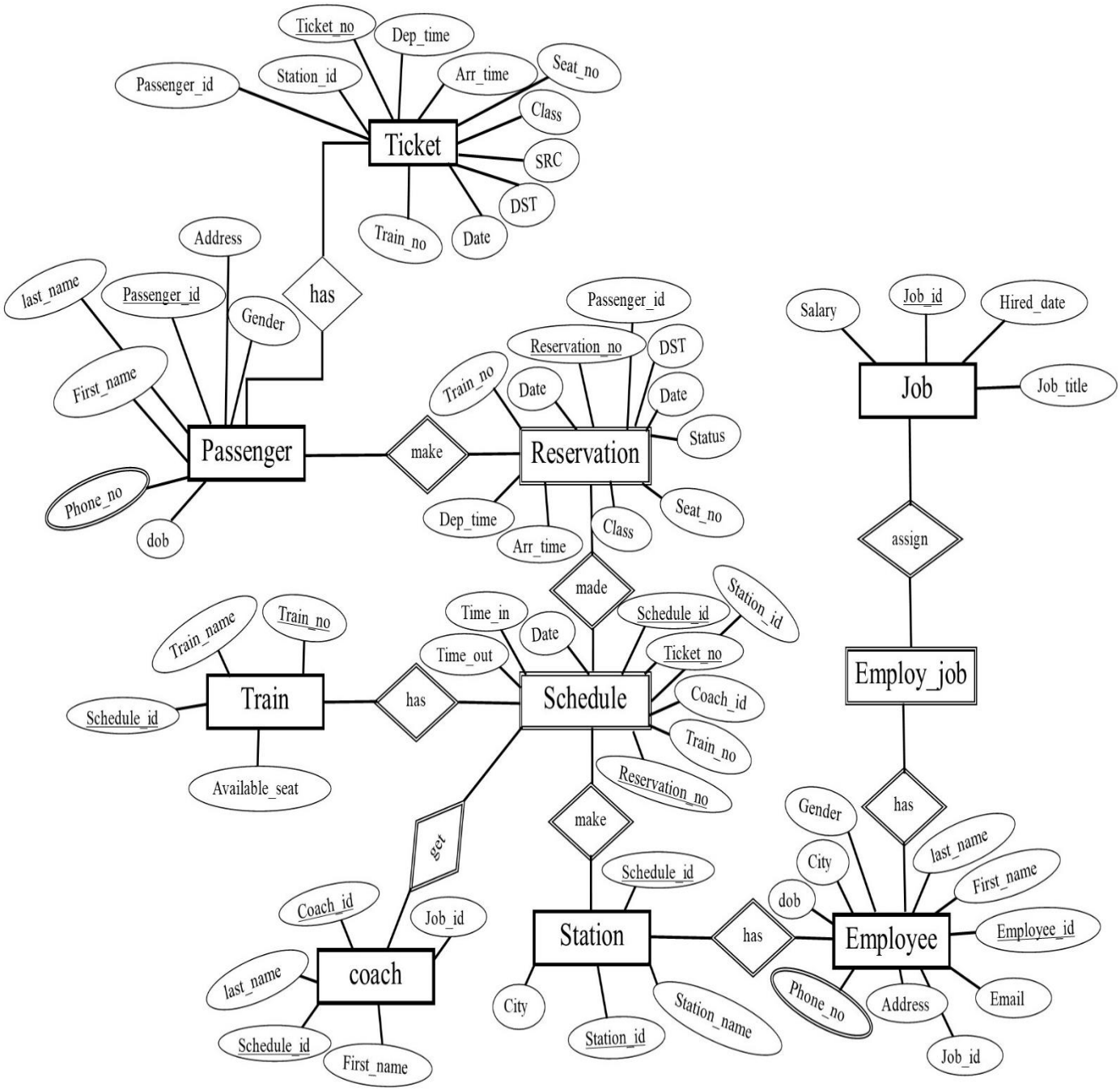


Figure 1: *ER diagram*

3- RELATIONAL MODEL

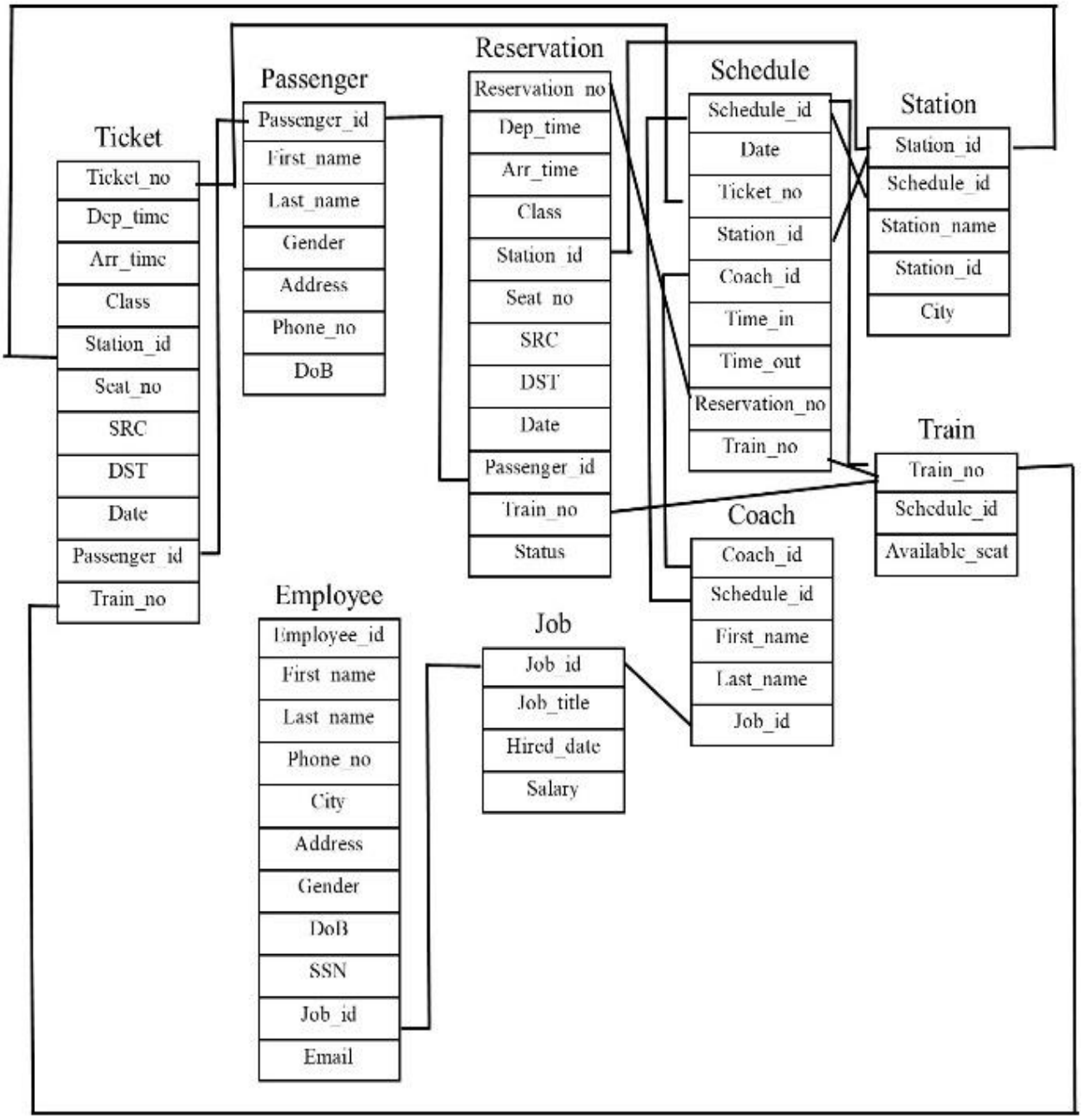
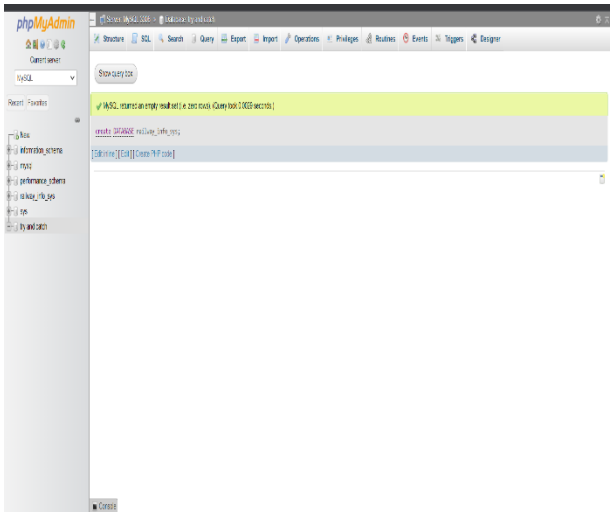
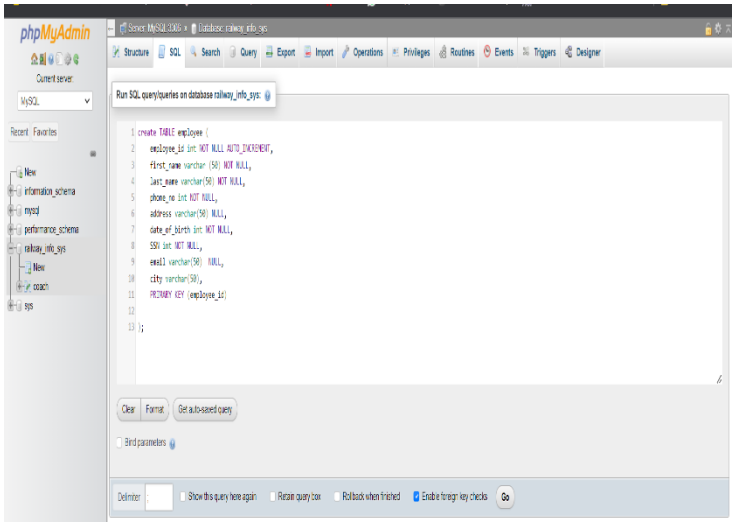


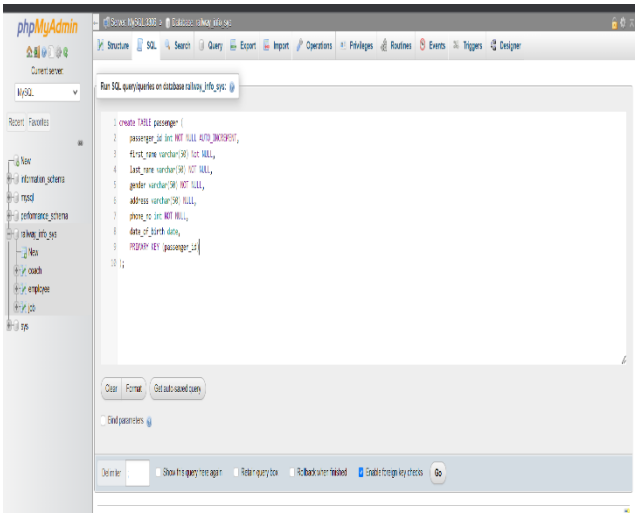
Figure 2: *Relational Model*

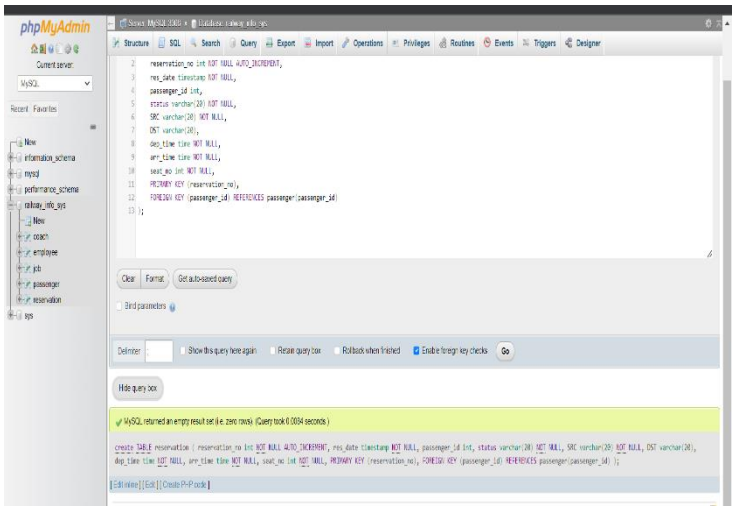
4- DATA DEFINITION LANGUAGE (DDL)

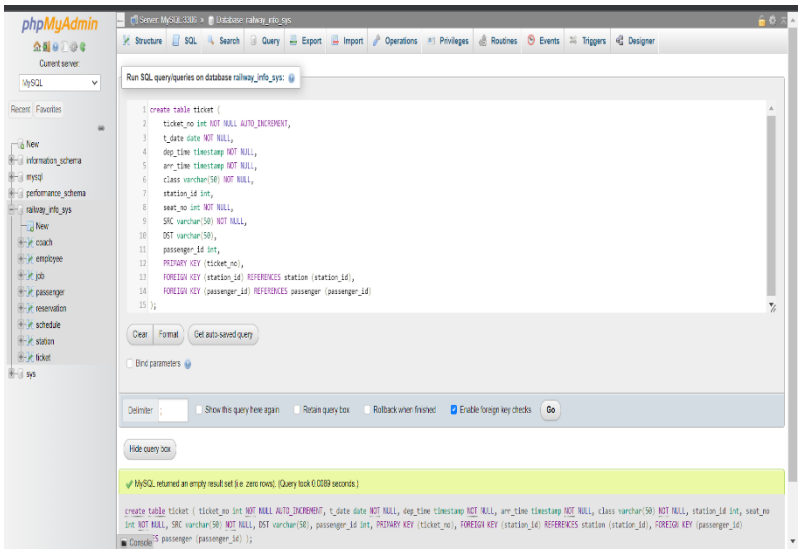
Creation of Database and few table schemas;





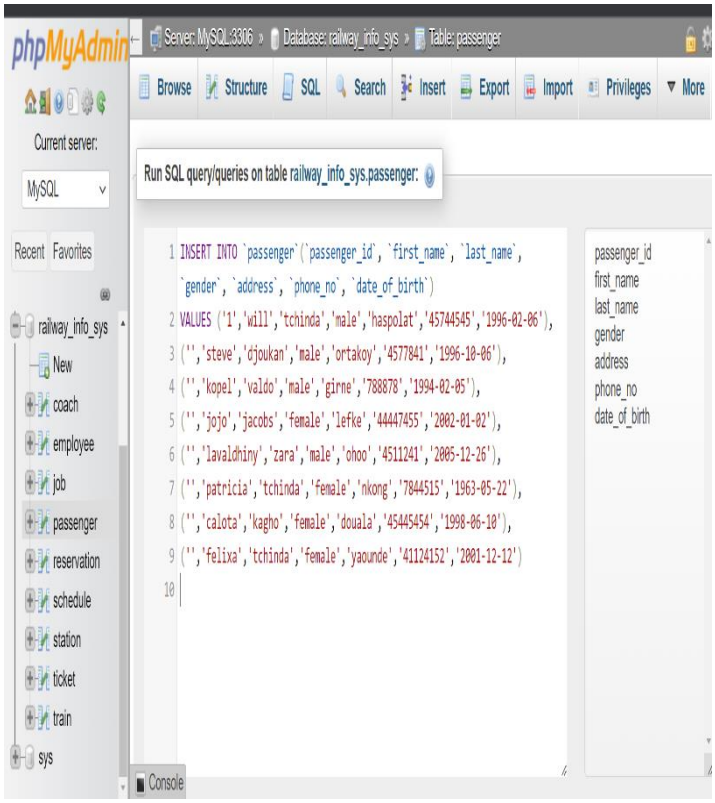


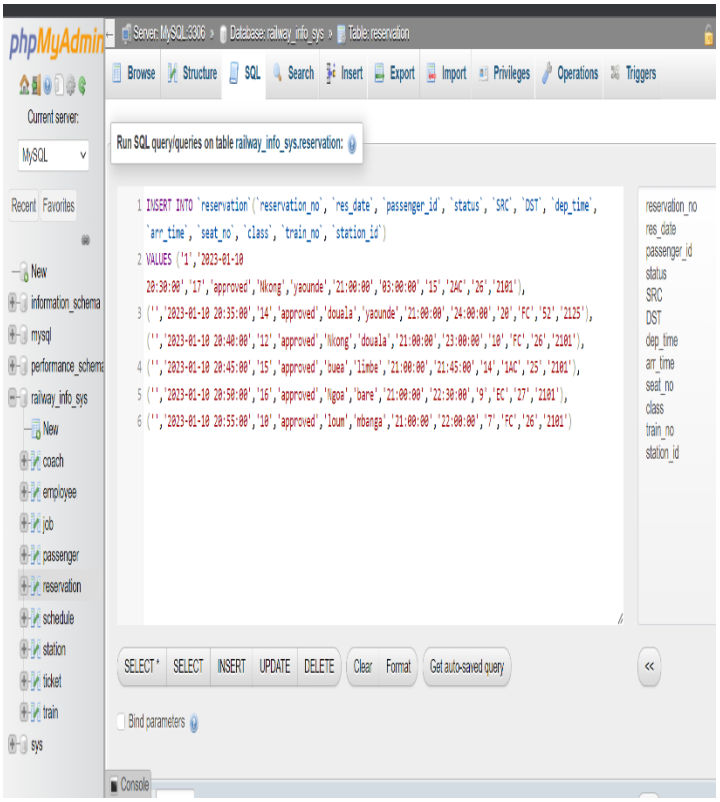


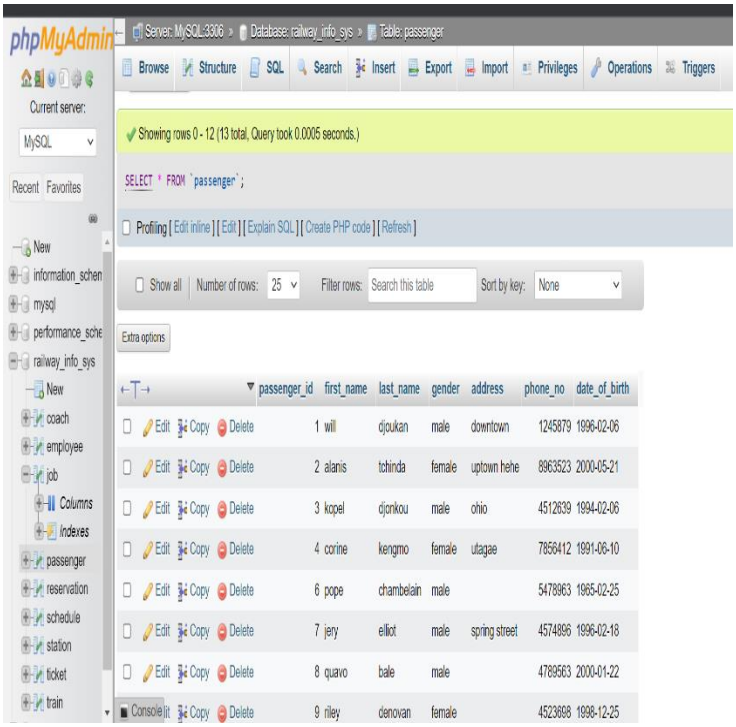


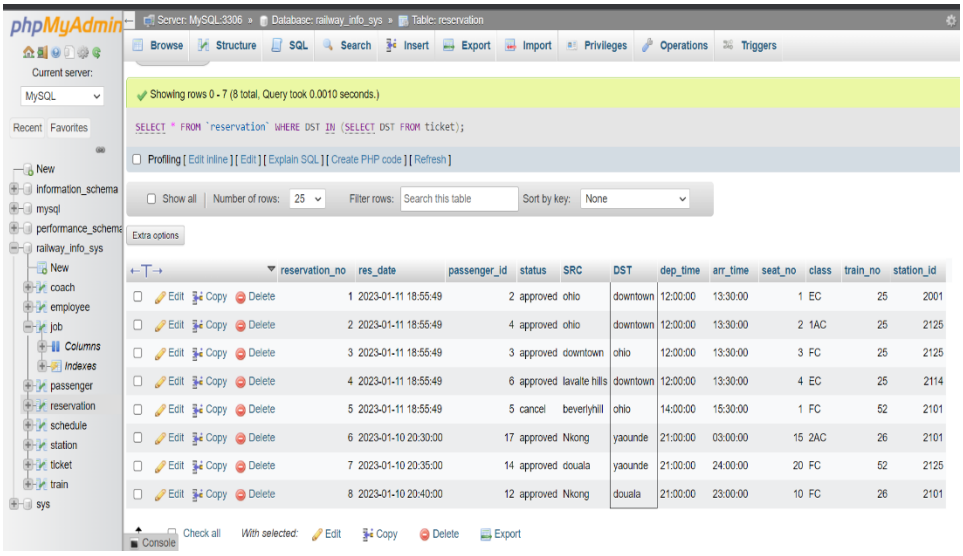
5- DATA MANIPULATION LANGUAGE (DML)

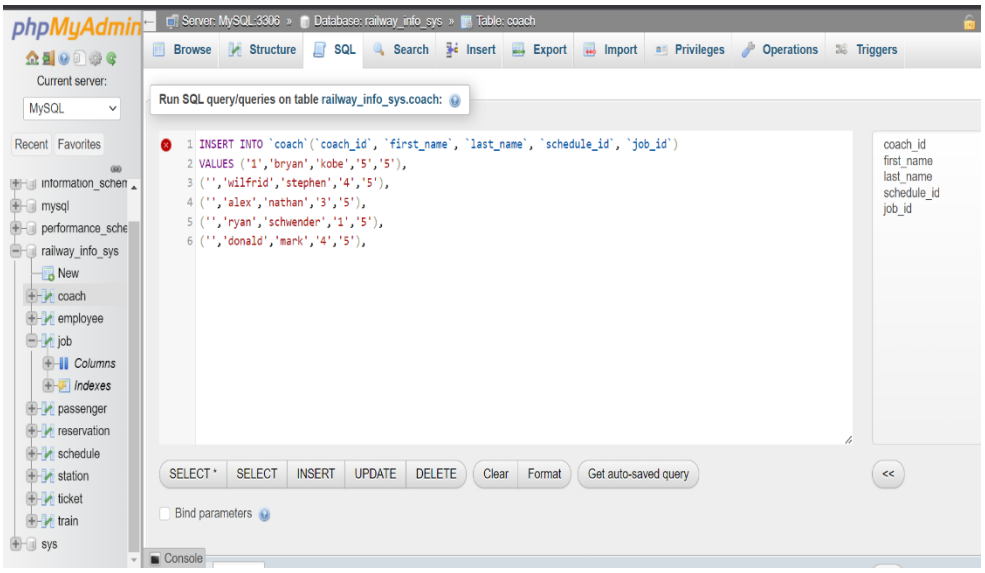
This section shows the different kind of data manipulation in the database;

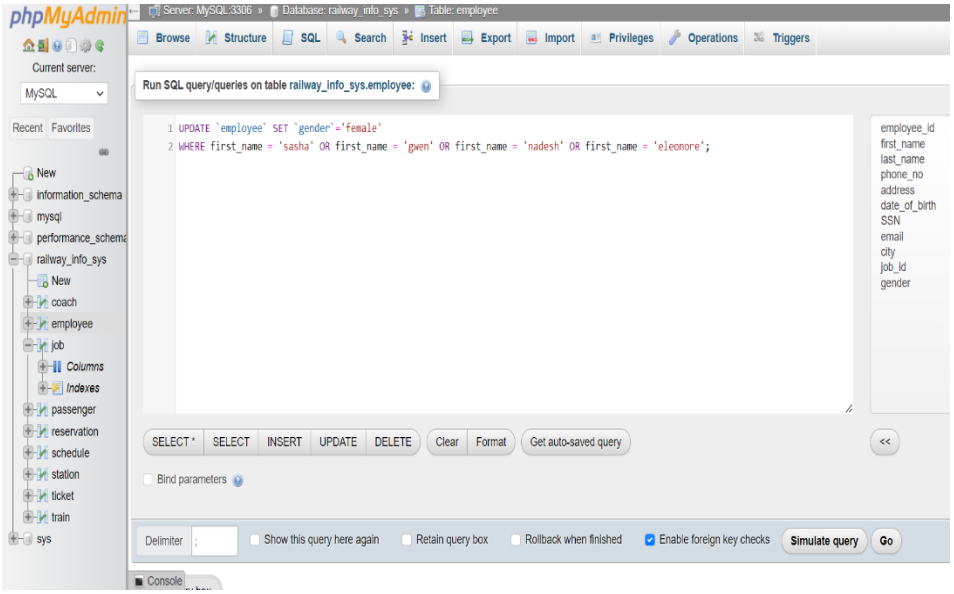







phpMyAdmin
Server: MySQL:3306 » Database: railway_info_sys » Table: reservation
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Showing rows 0 - 7 (8 total, Query took 0.0010 seconds.)
SELECT * FROM `reservation` WHERE DST IN (SELECT DST FROM ticket);
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
reservation_no res_date passenger_id status SRC DST dep_time arr_time seat_no class train_no station_id
1 2023-01-11 18:55:49 2 approved ohio downtown 12:00:00 13:30:00 1 EC 25 2001
2 2023-01-11 18:55:49 4 approved ohio downtown 12:00:00 13:30:00 2 1AC 25 2125
3 2023-01-11 18:55:49 3 approved downtown ohio 12:00:00 13:30:00 3 FC 25 2125
4 2023-01-11 18:55:49 6 approved lavalte hills downtown 12:00:00 13:30:00 4 EC 25 2114
5 2023-01-11 18:55:49 5 cancel beverlyhill ohio 14:00:00 15:30:00 1 FC 52 2101
6 2023-01-10 20:30:00 17 approved Nkong yaounde 21:00:00 03:00:00 15 2AC 26 2101
7 2023-01-10 20:35:00 14 approved douala yaounde 21:00:00 24:00:00 20 FC 52 2125
8 2023-01-10 20:40:00 12 approved Nkong douala 21:00:00 23:00:00 10 FC 26 2101
Check all With selected: Edit Copy Delete Export
Console


phpMyAdmin
Server: MySQL:3306 » Database: railway_info_sys » Table: coach
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Run SQL query/queries on table railway_info_sys.coach:
INSERT INTO `coach`(`coach_id`, `first_name`, `last_name`, `schedule_id`, `job_id`)
VALUES ('1','bryan','kobe','5','5'),
('','wilfrid','stephen','4','5'),
('','alex','nathan','3','5'),
('','ryan','schwender','1','5'),
('','donald','mark','4','5'),
coach_id
first_name
last_name
schedule_id
job_id
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
Bind parameters
Console


phpMyAdmin
Server: MySQL:3306 » Database: railway_info_sys » Table: employee
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Run SQL query/queries on table railway_info_sys.employee:
UPDATE `employee` SET `gender`='female'
WHERE first_name = 'sasha' OR first_name = 'gwen' OR first_name = 'nadesh' OR first_name = 'eleonore';
employee_id
first_name
last_name
phone_no
address
date_of_birth
SSN
email
city
job_id
gender
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
Bind parameters
Delimiter ; Show this query here again Retain query box Rollback when finished Enable foreign key checks Simulate query Go
Console

5.1 -- Database: `railway_info_sys`

```
-- Table structure for table `coach`

CREATE TABLE IF NOT EXISTS `coach` (
  `coach_id` int NOT NULL AUTO_INCREMENT,
  `first_name` varchar(50) NOT NULL,
  `last_name` varchar(50) NOT NULL,
  `schedule_id` int DEFAULT NULL,
  `job_id` int NOT NULL,
  PRIMARY KEY (`coach_id`),
  KEY `job_id` (`job_id`),
  KEY `schedule_id` (`schedule_id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `coach` (`coach_id`, `first_name`, `last_name`, `schedule_id`, `job_id`) VALUES
(1, 'john', 'kaminga', 2, 5),
(2, 'ricky', 'dave', 1, 5),
(3, 'rigobert', 'josh', 4, 5),
(4, 'gerald', 'peter', 3, 5);

-- Constraints for table `coach`

ALTER TABLE `coach`
  ADD CONSTRAINT `coach_ibfk_1` FOREIGN KEY (`job_id`) REFERENCES `job` (`job_id`) ON
DELETE RESTRICT ON UPDATE RESTRICT,
  ADD CONSTRAINT `coach_ibfk_2` FOREIGN KEY (`schedule_id`) REFERENCES `schedule`
(`schedule_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
COMMIT;

-- Table structure for table `employee`

CREATE TABLE IF NOT EXISTS `employee` (
  `employee_id` int NOT NULL AUTO_INCREMENT,
  `first_name` varchar(50) NOT NULL DEFAULT 'NOT NULL',
  `last_name` varchar(50) NOT NULL DEFAULT 'NOT NULL',
  `phone_no` int NOT NULL,
  `date_of_birth` date NOT NULL,
  `gender` varchar(10) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL
DEFAULT 'NOT NULL',
  `SSN` int NOT NULL,
  `email` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL,
  `city` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL,
  `address` varchar(50) DEFAULT NULL,
  `job_id` int DEFAULT NULL,
  PRIMARY KEY (`employee_id`),
  KEY `employee_ibfk_1` (`job_id`)
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

```
INSERT INTO `employee` (`employee_id`, `first_name`, `last_name`, `phone_no`, `date_of_birth`,
`gender`, `SSN`, `email`, `city`, `address`, `job_id`) VALUES
(1, 'sasha', 'maeve', 2564574, '1963-10-20', 'female', 784, 'que@gmailcom', 'ohio', 'dalington street', 2),
(2, 'gwen', 'gwei', 45477454, '1992-02-06', 'female', 546, '', 'douala', 'cite verte', 8),
(3, 'calid', 'khal', 4789562, '1975-06-20', 'male', 545, 'bew@gmailcom', 'beverlyhill', 'hillhigh street', 1),
(4, 'caleb', 'antoine', 1254789, '1985-05-12', 'male', 778, 'ash@gmailcom', 'orgen', 'hallway street', 4),
(5, 'nadesh', 'wiley', 548966, '1988-12-12', 'female', 565, 'wal@gmailcom', 'quatemala', 'queens street', 3),
(6, 'rosine', 'bouda', 45122145, '1989-12-11', 'female', 112, '', 'douala', 'toiture rouge', 7),
(7, 'eleonore', 'tchang', 745445, '1973-11-25', 'female', 331, '', 'nkong', 'rond point', 6);

-- Constraints for table `employee`

ALTER TABLE `employee`
  ADD CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`job_id`) REFERENCES `job` (`job_id`)
ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;

-- Table structure for table `job`

CREATE TABLE IF NOT EXISTS `job` (
  `job_id` int NOT NULL AUTO_INCREMENT,
  `job_title` varchar(50) NOT NULL,
  `hired_date` date NOT NULL,
  `salary` int NOT NULL,
  PRIMARY KEY (`job_id`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `job` (`job_id`, `job_title`, `hired_date`, `salary`) VALUES
(1, 'ticket controler', '2012-10-12', 70000),
(2, 'cashier', '2018-01-20', 80000),
(3, 'head officer', '2010-05-02', 350000),
(4, 'station agent', '2015-08-15', 75000),
(5, 'coach', '2010-11-30', 85000),
(6, 'station mistress', '2001-06-25', 500000),
(7, 'accountant', '2005-10-05', 250000),
(8, 'receptionist', '2010-01-15', 50000);
COMMIT;

-- Table structure for table `passenger`

CREATE TABLE IF NOT EXISTS `passenger` (
  `passenger_id` int NOT NULL AUTO_INCREMENT,
  `first_name` varchar(50) NOT NULL DEFAULT 'NOT NULL',
  `last_name` varchar(50) NOT NULL DEFAULT 'NOT NULL',
  `date_of_birth` date DEFAULT NULL,
  `gender` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  `phone_no` int NOT NULL,
  `address` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL
DEFAULT 'NOT NULL',
```

```
  PRIMARY KEY (`passenger_id`)
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `passenger` (`passenger_id`, `first_name`, `last_name`, `date_of_birth`, `gender`,
`phone_no`, `address`) VALUES
(1, 'will', 'djoukan', '1996-02-06', 'male', 1245879, 'downtown'),
(2, 'alanis', 'tchinda', '2000-05-21', 'female', 8963523, 'uptown hehe'),
(3, 'kopel', 'djonkou', '1994-02-06', 'male', 4512639, 'ohio'),
(4, 'corine', 'kengmo', '1991-06-10', 'female', 7856412, 'utagae'),
(5, 'pope', 'chambelain', '1965-02-25', 'male', 5478963, 'quatemala'),
(6, 'jerry', 'elliot', '1996-02-18', 'male', 4574896, 'spring street'),
(7, 'quavo', 'bale', '2000-01-22', 'male', 4789563, 'beverlyhill'),
(8, 'riley', 'denovan', '1998-12-25', 'female', 4523698, 'orgen'),
(9, 'steve', 'djoukan', '1996-10-06', 'male', 4577841, 'ortakoy'),
(10, 'lavaldhiny', 'zara', '2005-12-26', 'male', 4511241, 'ohoo'),
(11, 'patricia', 'tchinda', '1963-05-22', 'female', 7844515, 'nkong'),
(12, 'calota', 'kagho', '1998-06-10', 'female', 45445454, 'douala'),
(13, 'felixa', 'tchinda', '2001-12-12', 'female', 41124152, 'yaounde');
COMMIT;

-- Table structure for table `reservation`

CREATE TABLE IF NOT EXISTS `reservation` (
  `reservation_no` int NOT NULL AUTO_INCREMENT,
  `res_date` timestamp NOT NULL,
  `passenger_id` int NOT NULL,
  `status` varchar(20) NOT NULL,
  `SRC` varchar(20) NOT NULL,
  `DST` varchar(20) DEFAULT NULL,
  `dep_time` time NOT NULL,
  `arr_time` time NOT NULL,
  `seat_no` int NOT NULL,
  `class` varchar(50) DEFAULT NULL,
  `train_no` int DEFAULT NULL,
  `station_id` int NOT NULL,
  PRIMARY KEY (`reservation_no`),
  KEY `station_id` (`station_id`),
  KEY `train_no` (`train_no`),
  KEY `passenger_id` (`passenger_id`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `reservation` (`reservation_no`, `res_date`, `passenger_id`, `status`, `SRC`, `DST`,
`dep_time`, `arr_time`, `seat_no`, `class`, `train_no`, `station_id`) VALUES
(1, '2023-01-11 16:55:49', 2, 'approved', 'ohio', 'downtown', '12:00:00', '13:30:00', 1, 'EC', 25, 2001),
(2, '2023-01-11 16:55:49', 4, 'approved', 'ohio', 'downtown', '12:00:00', '13:30:00', 2, '1AC', 25, 2125),
(3, '2023-01-11 16:55:49', 3, 'approved', 'downtown', 'ohio', '12:00:00', '13:30:00', 3, 'FC', 25, 2125),
(4, '2023-01-11 16:55:49', 6, 'approved', 'lavalte hills', 'downtown', '12:00:00', '13:30:00', 4, 'EC', 25,
2114),
(5, '2023-01-11 16:55:49', 5, 'cancel', 'beverlyhill', 'ohio', '14:00:00', '15:30:00', 1, 'FC', 52, 2101),
```

```
(6, '2023-01-10 18:30:00', 11, 'approved', 'Nkong', 'yaounde', '21:00:00', '03:00:00', 15, '2AC', 26, 2101),
(7, '2023-01-10 18:35:00', 8, 'approved', 'douala', 'yaounde', '21:00:00', '24:00:00', 20, 'FC', 52, 2125),
(8, '2023-01-10 18:40:00', 12, 'approved', 'Nkong', 'douala', '21:00:00', '23:00:00', 10, 'FC', 26, 2101),
(9, '2023-01-10 18:45:00', 7, 'approved', 'buea', 'limbe', '21:00:00', '21:45:00', 14, '1AC', 25, 2101),
(10, '2023-01-10 18:50:00', 1, 'approved', 'Ngoa', 'bare', '21:00:00', '22:30:00', 9, 'EC', 27, 2101),
(11, '2023-01-10 18:55:00', 10, 'approved', 'loum', 'mbanga', '21:00:00', '22:00:00', 7, 'FC', 26, 2101);

-- Constraints for table `reservation`

ALTER TABLE `reservation`
  ADD CONSTRAINT `reservation_ibfk_1` FOREIGN KEY (`station_id`) REFERENCES `station`
(`station_id`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  ADD CONSTRAINT `reservation_ibfk_2` FOREIGN KEY (`train_no`) REFERENCES `train`
(`train_no`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  ADD CONSTRAINT `reservation_ibfk_3` FOREIGN KEY (`passenger_id`) REFERENCES
`passenger` (`passenger_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
COMMIT;

-- Table structure for table `schedule`

CREATE TABLE IF NOT EXISTS `schedule` (
  `schedule_id` int NOT NULL AUTO_INCREMENT,
  `time_in` time NOT NULL,
  `time_out` time NOT NULL,
  `coach_id` int DEFAULT NULL,
  `reservation_id` int DEFAULT NULL,
  `ticket_no` int DEFAULT NULL,
  `train_no` int DEFAULT NULL,
  `station_id` int DEFAULT NULL,
  PRIMARY KEY (`schedule_id`),
  KEY `coach_id` (`coach_id`),
  KEY `reservation_id` (`reservation_id`),
  KEY `ticket_no` (`ticket_no`),
  KEY `train_no` (`train_no`),
  KEY `station_id` (`station_id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `schedule` (`schedule_id`, `time_in`, `time_out`, `coach_id`, `reservation_id`, `ticket_no`,
`train_no`, `station_id`) VALUES
(1, '15:50:00', '16:00:00', 3, 5, 1001, 25, 2001),
(2, '13:30:00', '12:00:00', 1, 2, 1005, 26, 2114),
(3, '15:30:00', '14:00:00', 1, 1, 1003, 52, 2114),
(4, '15:50:00', '14:50:00', 3, 3, 1004, 25, 2125),
(5, '13:30:00', '12:00:00', 3, 4, 1002, 25, 2101);

-- Constraints for table `schedule`

ALTER TABLE `schedule`
  ADD CONSTRAINT `schedule_ibfk_1` FOREIGN KEY (`coach_id`) REFERENCES `coach`
(`coach_id`) ON DELETE RESTRICT ON UPDATE RESTRICT,
```

```
  ADD CONSTRAINT `schedule_ibfk_2` FOREIGN KEY (`reservation_id`) REFERENCES
`reservation` (`reservation_no`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  ADD CONSTRAINT `schedule_ibfk_3` FOREIGN KEY (`ticket_no`) REFERENCES `ticket`
(`ticket_no`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  ADD CONSTRAINT `schedule_ibfk_4` FOREIGN KEY (`train_no`) REFERENCES `train`
(`train_no`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  ADD CONSTRAINT `schedule_ibfk_5` FOREIGN KEY (`station_id`) REFERENCES `station`
(`station_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
COMMIT;

-- Table structure for table `station`

CREATE TABLE IF NOT EXISTS `station` (
  `station_id` int NOT NULL,
  `station_name` varchar(50) NOT NULL,
  `city` varchar(50) NOT NULL,
  `schedule_id` int DEFAULT NULL,
  PRIMARY KEY (`station_id`),
  KEY `schedule_id` (`schedule_id`),
  KEY `station_id` (`station_id`),
  KEY `station_id_2` (`station_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `station` (`station_id`, `station_name`, `city`, `schedule_id`) VALUES
(2001, 'Westbrone', 'ohio', 5),
(2101, 'Bronx', 'downtown', 1),
(2114, 'Trinity', 'beverlyhill', 2),
(2120, 'Dracant', 'treyo', 3),
(2125, 'downside', 'califort', 5),
(2126, 'Stark', 'downtown', 4);

-- Constraints for table `station`

ALTER TABLE `station`
  ADD CONSTRAINT `station_ibfk_1` FOREIGN KEY (`schedule_id`) REFERENCES `schedule`
(`schedule_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
COMMIT;

-- Table structure for table `ticket`

CREATE TABLE IF NOT EXISTS `ticket` (
  `ticket_no` int NOT NULL AUTO_INCREMENT,
  `t_date` datetime NOT NULL,
  `dep_time` time NOT NULL,
  `arr_time` time NOT NULL,
  `class` varchar(50) NOT NULL,
  `station_id` int DEFAULT NULL,
  `seat_no` int NOT NULL,
  `SRC` varchar(50) NOT NULL,
  `DST` varchar(50) DEFAULT NULL,
  `passenger_id` int DEFAULT NULL,
```

```
  `train_no` int DEFAULT NULL,
  PRIMARY KEY (`ticket_no`),
  KEY `passenger_id` (`passenger_id`),
  KEY `station_id` (`station_id`),
  KEY `train_no` (`train_no`)
) ENGINE=InnoDB AUTO_INCREMENT=1006 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `ticket` (`ticket_no`, `t_date`, `dep_time`, `arr_time`, `class`, `station_id`, `seat_no`,
`SRC`, `DST`, `passenger_id`, `train_no`) VALUES
(1001, '2023-01-11 14:00:00', '14:50:00', '15:50:48', 'EC', 2001, 8, 'oweho', 'ohio', 8, 26),
(1002, '2023-01-11 14:05:00', '14:50:00', '16:20:00', 'EC', 2120, 9, 'treyo', 'califort', 7, 25),
(1003, '2023-01-11 14:30:00', '14:50:00', '15:30:00', '2AC', 2101, 7, 'tradot', 'downtown', 1, 25),
(1004, '2023-01-11 14:40:00', '14:50:00', '16:20:00', 'FC', 2125, 1, 'treyo', 'yaounde', 9, 25),
(1005, '2023-01-11 14:40:00', '16:50:00', '17:20:00', 'FC', 2120, 6, 'queens', 'douala', 10, 26);

-- Constraints for table `ticket`

ALTER TABLE `ticket`
  ADD CONSTRAINT `ticket_ibfk_1` FOREIGN KEY (`passenger_id`) REFERENCES `passenger`
(`passenger_id`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  ADD CONSTRAINT `ticket_ibfk_2` FOREIGN KEY (`station_id`) REFERENCES `station`
(`station_id`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  ADD CONSTRAINT `ticket_ibfk_3` FOREIGN KEY (`train_no`) REFERENCES `train` (`train_no`)
ON DELETE RESTRICT ON UPDATE RESTRICT;
COMMIT;

-- Table structure for table `train`

DROP TABLE IF EXISTS `train`;
CREATE TABLE IF NOT EXISTS `train` (
  `train_no` int NOT NULL,
  `available_seat` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT
NULL,
  `schedule_id` int DEFAULT NULL,
  PRIMARY KEY (`train_no`),
  KEY `schedule_id` (`schedule_id`),
  KEY `train_no` (`train_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `train` (`train_no`, `available_seat`, `schedule_id`) VALUES
(25, '5', 1),
(26, '0', 2),
(27, '25', 4),
(52, '2', 3);

-- Constraints for table `train`
ALTER TABLE `train`
  ADD CONSTRAINT `train_ibfk_1` FOREIGN KEY (`schedule_id`) REFERENCES `schedule`
(`schedule_id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
COMMIT;
```

6- NORMALIZATION

The database was been evaluated through the normalization technique to find out any redundancy while inserting, updating and deleting data in the database. We analyzed the data through the most common normal forms used. They are;

1- **First Normal Form (1NF):**

   In here, all our cells contained exactly one value for each attribute define in the database.

2- **Second Normal Form (2NF):**

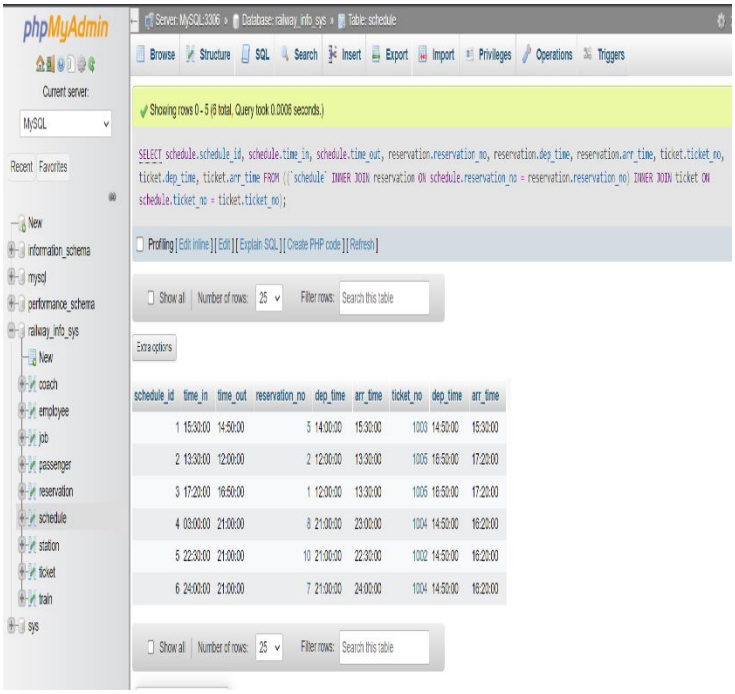
   In here, we didn't have any partial dependency since every non-key column is dependent upon its **PRIMAY KEY**.

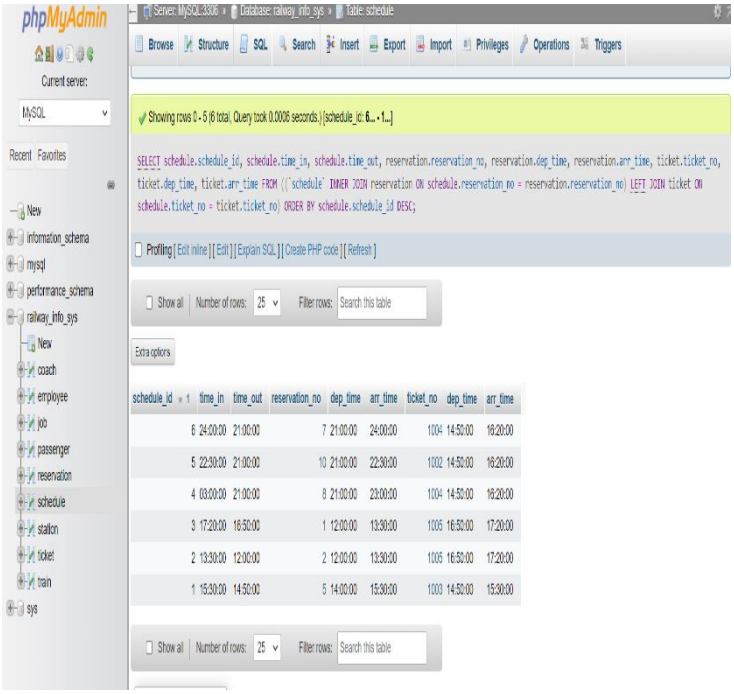
3- **Third Normal Form (3NF):**

   In here, we didn't encountered any transitive dependency in the tables.

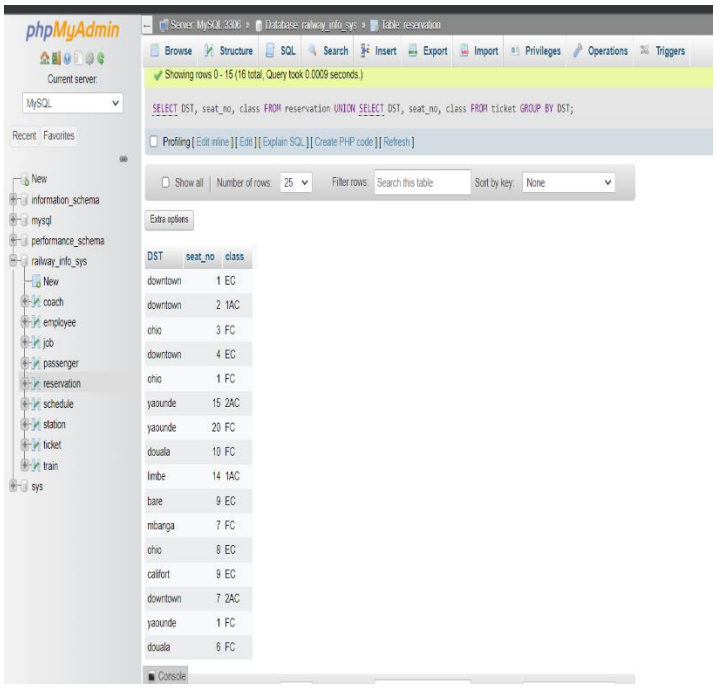
7- RETRIEVAL INFORMATION FOR MANAGEMENT

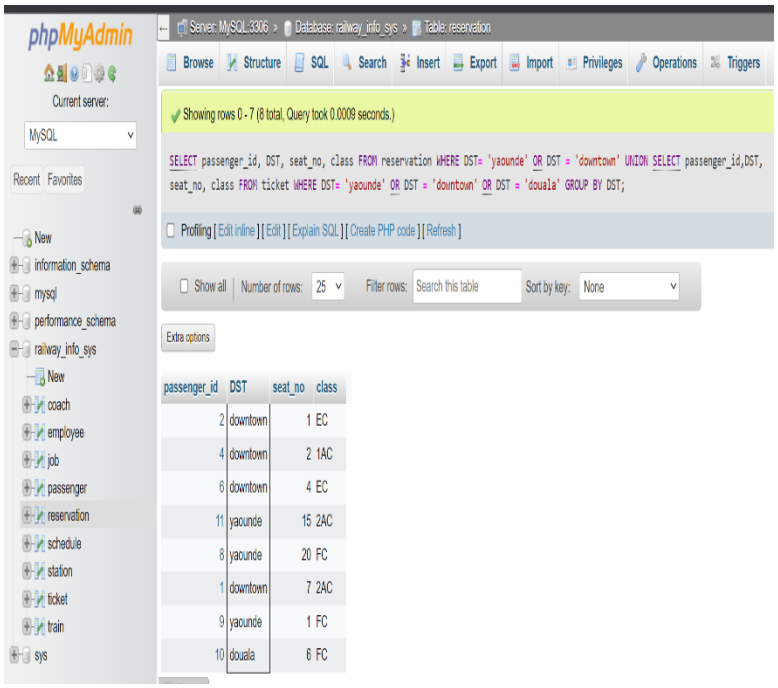
1. The query below shows the time a train passes through a station based on a time in and time out schedule. In this query, I used inner join, left join Group By DESC;



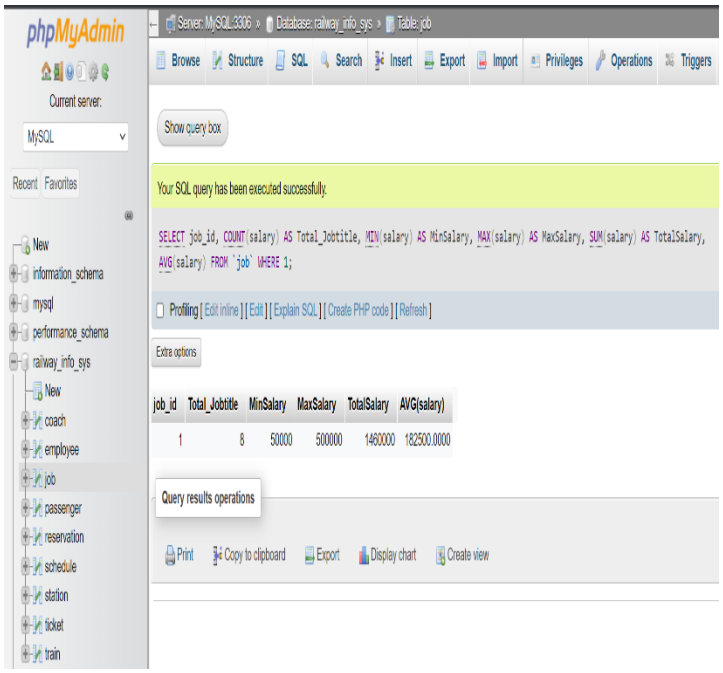


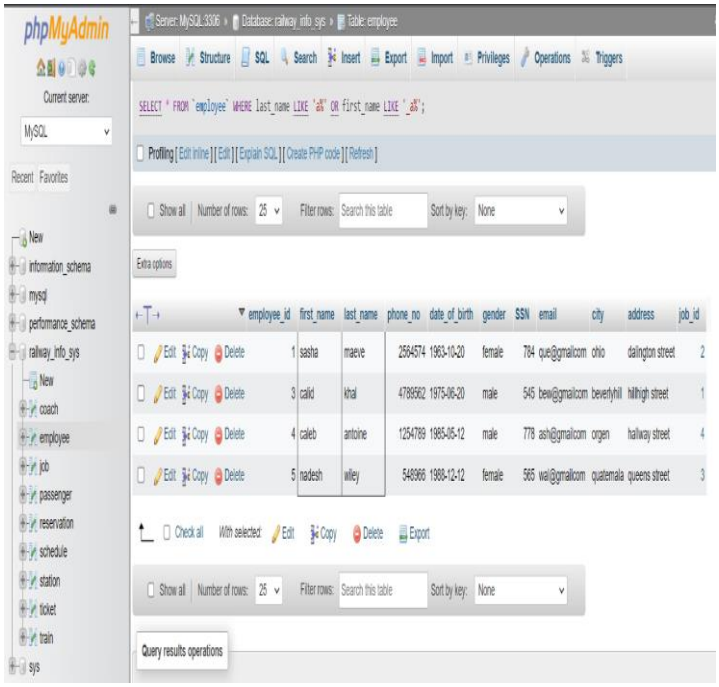
2. The query below shows a union of two different tables. In this query, I used Union operator and Where and Group By clauses;



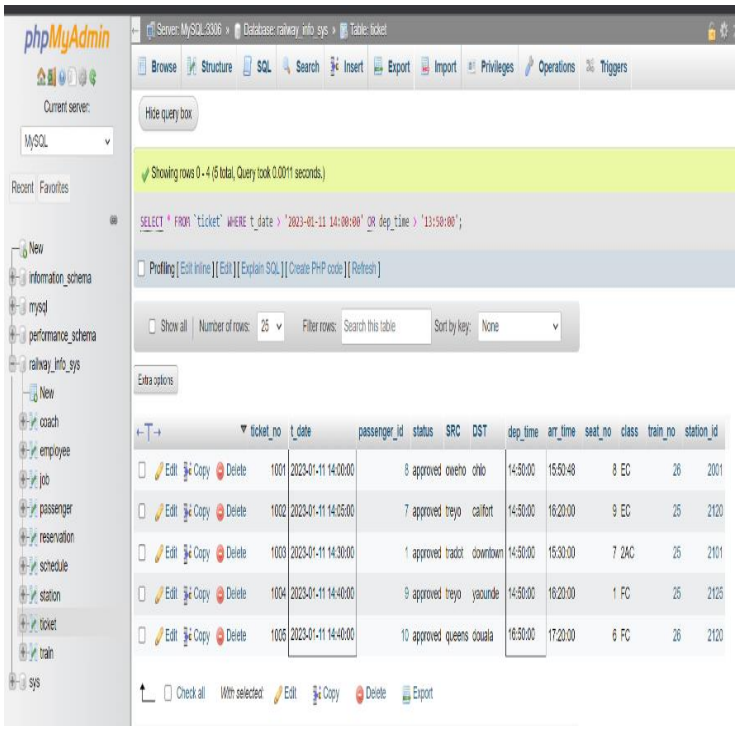


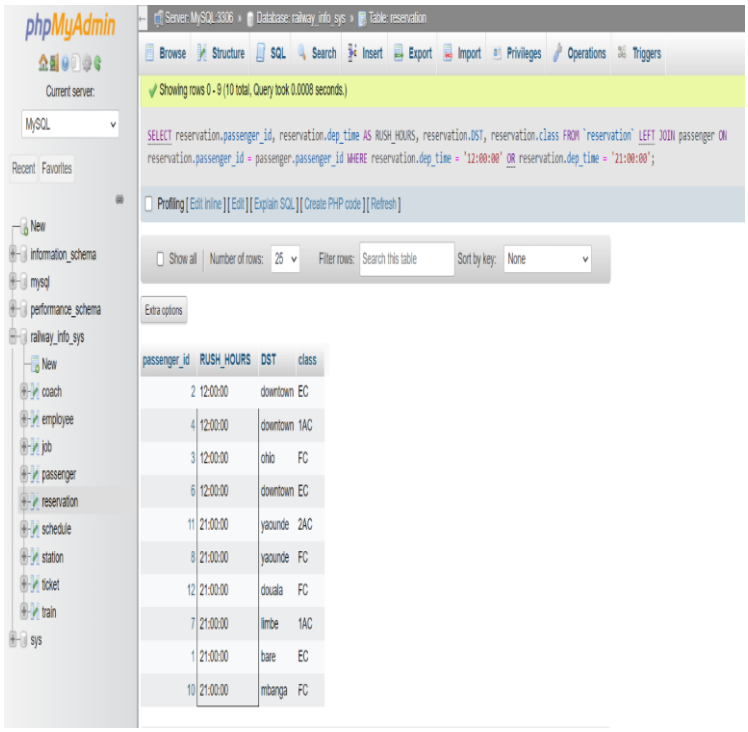
3. This query brings out the total number of job title, minimum, maximum and sum as well as average of the employee's salary. In this query, I also brought out all the employees name that first name started with "a" and the second letter was "a" using the wildcards character.



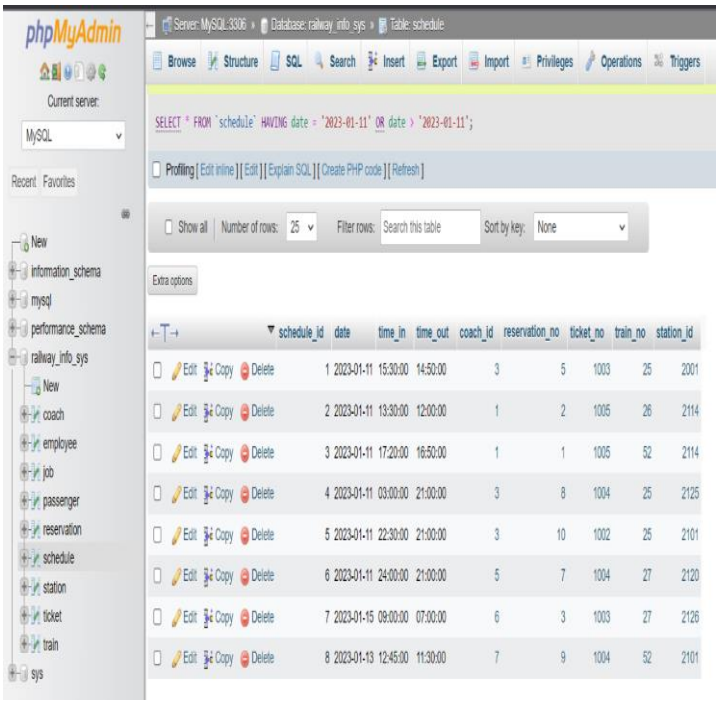


4. This query present time which is greater than a certain period. It also brings out the **rush hours** of the company. I used the WHERE and OR clauses.

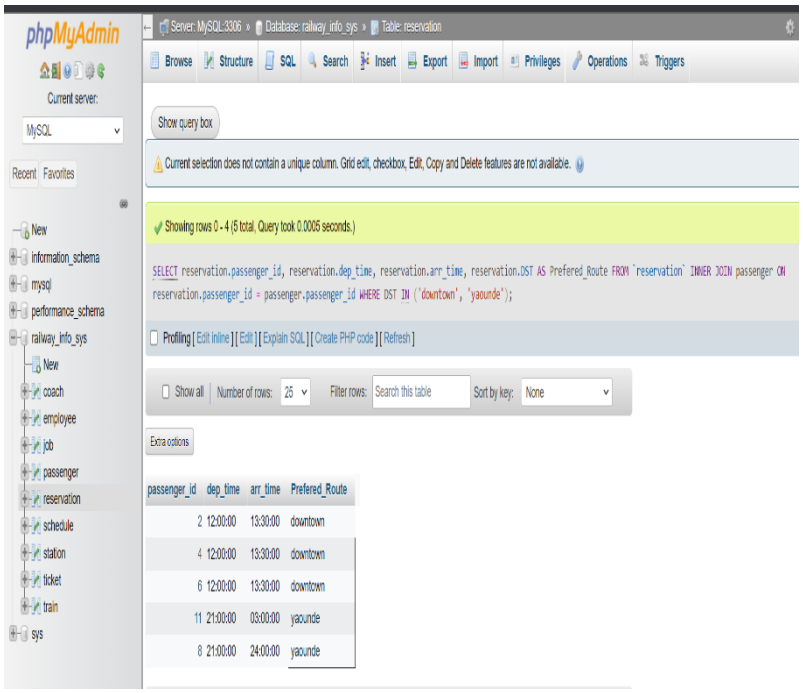




5. This query brings out the **busy schedule** of the company between other days. It also shows the preferred routes used by trains.

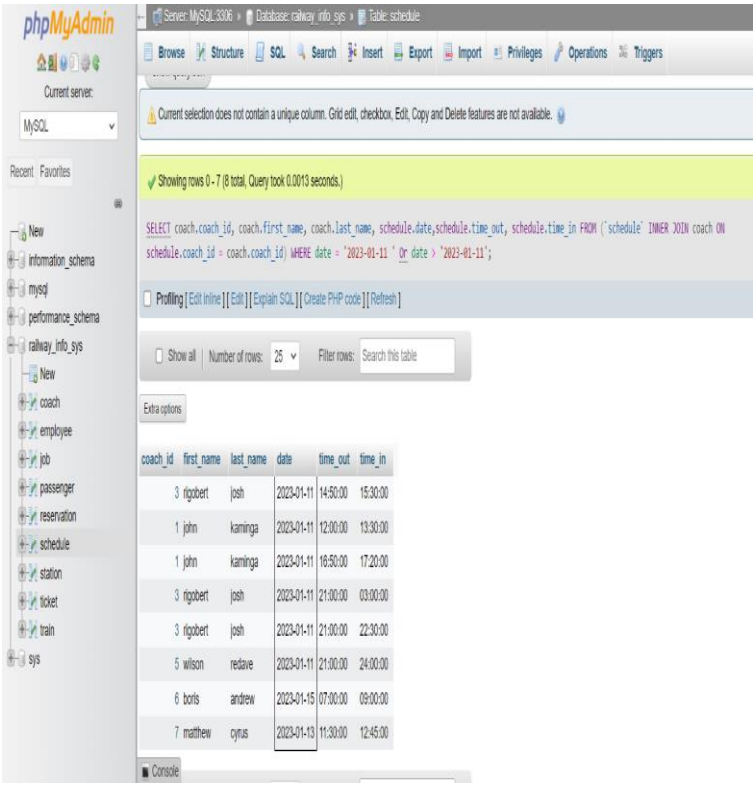


schedule_id	date	time_in	time_out	coach_id	reservation_no	ticket_no	train_no	station_id
1	2023-01-11	15:30:00	14:50:00	3	5	1003	25	2001
2	2023-01-11	13:30:00	12:00:00	1	2	1005	26	2114
3	2023-01-11	17:20:00	16:50:00	1	1	1005	52	2114
4	2023-01-11	03:00:00	21:00:00	3	8	1004	25	2125
5	2023-01-11	22:30:00	21:00:00	3	10	1002	25	2101
6	2023-01-11	24:00:00	21:00:00	5	7	1004	27	2120
7	2023-01-15	09:00:00	07:00:00	6	3	1003	27	2126
8	2023-01-13	12:45:00	11:30:00	7	9	1004	52	2101



passenger_id	dep_time	arr_time	Prefered_Route
2	12:00:00	13:30:00	downtown
4	12:00:00	13:30:00	downtown
6	12:00:00	13:30:00	downtown
11	21:00:00	03:00:00	yaounde
8	21:00:00	24:00:00	yaounde

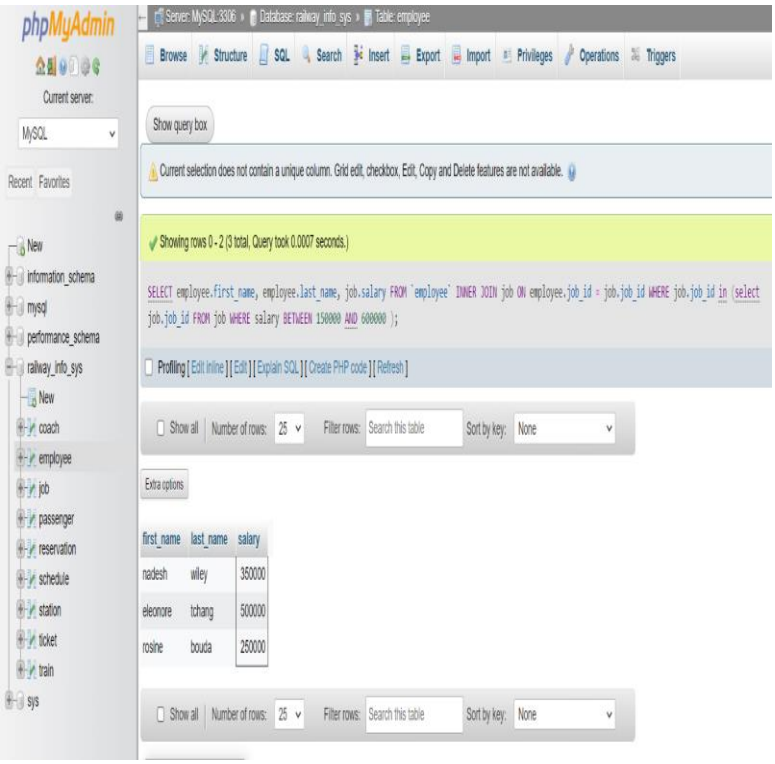
6. This query brings out the **coaches working hours, availability on a certain day and time**. Here, I used **subquery** and a **join** query to retrieve employee's salary.

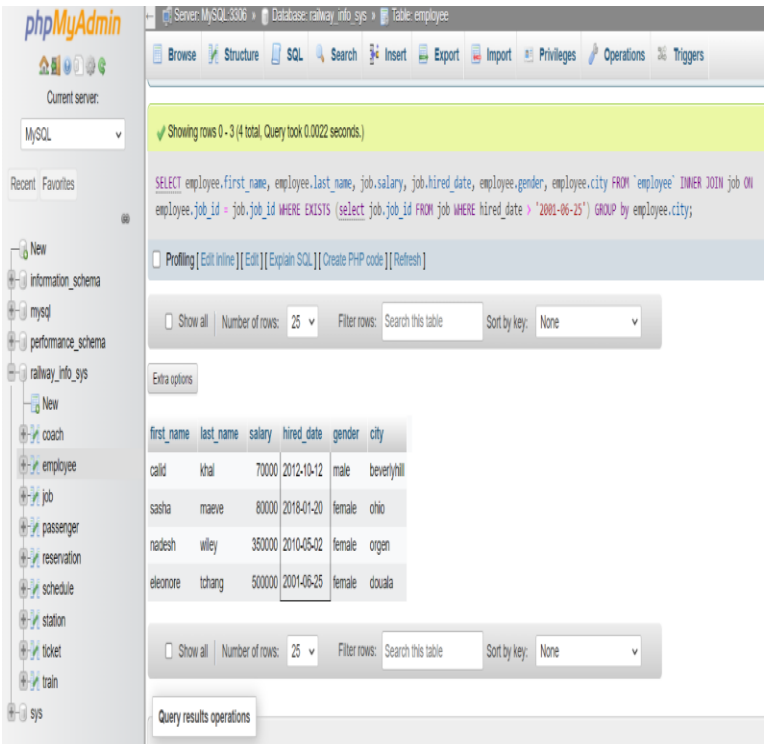


coach_id	first_name	last_name	date	time_out	time_in
3	rigobert	josh	2023-01-11	14:50:00	15:30:00
1	john	kaminga	2023-01-11	12:00:00	13:30:00
1	john	kaminga	2023-01-11	16:50:00	17:20:00
3	rigobert	josh	2023-01-11	21:00:00	03:00:00
3	rigobert	josh	2023-01-11	21:00:00	22:30:00
5	wilson	redave	2023-01-11	21:00:00	24:00:00
6	boris	andrew	2023-01-15	07:00:00	09:00:00
7	matthew	cyrus	2023-01-13	11:30:00	12:45:00

first_name	last_name	salary
calid	khal	70000
sasha	maeve	80000
nadesh	wiley	350000
caleb	antoine	75000
eleonore	tchang	500000
rosine	bouda	250000
gwen	gwel	50000

7. This query brings out the salary of employees between certain ranges by using the **BETWEEN** and **AND** clauses. Here, I also used the Exist operator to find the existence of records.





8. This query retrieve the schedule of a given date and time which presents the passenger_id, coach name, DST, train_no and station_name by using subqueries and join queries.