

COMS2002A: DATABASE FUNDAMENTALS

SEMESTER 1, 2023

LAB 5 – PRIMARY KEYS, FOREIGN KEYS, JOINS

5th April, 2023

Lecturer: Dr. Olaperi Okuboyejo

PRELIMINARIES

1. Connect to the LAMP server
2. Connect to MySQL
3. Choose the database to be used using the `USE` command.

Details on these steps are in the **Lab1.pdf** file.

PART 1 – PRACTICE

The lab builds on previous class lectures, especially from Unit 3.

For more details on the SQL syntax used in this lab, visit the course resource -
Introduction to SQL – 5.pdf

A. CREATING TABLES WITH PRIMARY AND FOREIGN KEYS

1. You can add the primary key constraint to a table at the point of creation.

Let's create a table `VENDOR` that has attributes `VENDOR_NO` and `VENDOR_NAME`; `VENDOR_NO` being the primary key.

```
CREATE TABLE VENDOR (  
  VENDOR_NO INTEGER(11),  
  VENDOR_NAME VARCHAR(35),  
  PRIMARY KEY (VENDOR_NO));
```

After specifying table columns and their constraints (if any), we specify the columns selected as the primary key, using the PRIMARY KEY command and the columns in brackets. Remember that you can have more than one attribute as the primary key.

2. Next, describe the table to see the details. Notice that the `VENDOR_NO` attribute is marked as the primary key (PRI).

```
mysql> DESC VENDOR;
+-----+-----+-----+-----+-----+-----+
| Field      | Type        | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| VENDOR_NO  | int         | NO   | PRI | NULL    |      |
| VENDOR_NAME | varchar(35) | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

3. You can also specify a primary key in an already existing table using the ALTER command. Let's create a similar table called `VENDOR_NEW`. This time without specifying the primary key.

```
CREATE TABLE VENDOR_NEW (
VENDOR_NO INTEGER(11),
VENDOR_NAME VARCHAR(35));
```

4. Describe the table to confirm that there is no key specified.

```
mysql> desc VENDOR_NEW;
+-----+-----+-----+-----+-----+-----+
| Field      | Type        | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| VENDOR_NO  | int         | YES  |     | NULL    |      |
| VENDOR_NAME | varchar(35) | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

5. Now, let's alter the table to add the PRIMARY KEY constraint.

```
ALTER TABLE VENDOR_NEW
ADD PRIMARY KEY (VENDOR_NO);
```

6. Describe the table again and observe the difference.

```
mysql> desc VENDOR_NEW;
+-----+-----+-----+-----+-----+-----+
| Field      | Type        | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| VENDOR_NO  | int         | NO   | PRI | NULL    |       |
| VENDOR_NAME | varchar(35) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

7. Note that you can also add a primary key constraint by using the ALTER TABLE MODIFY COLUMN command.
8. Next, let's add both a primary key and foreign key constraints at the point of table creation. Create a CUSTOMER table as follows:

```
CREATE TABLE CUSTOMER (
  CUSTOMER_ID INTEGER(5),
  CUSTOMER_NAME VARCHAR(35),
  VENDOR_NO INTEGER(11),
  PRIMARY KEY (CUSTOMER_ID),
  FOREIGN KEY (VENDOR_NO) REFERENCES
  VENDOR (VENDOR_NO) );
```

Notice that you specify the attribute that is a foreign key in a table. Then you specify the other table and attribute name that it references (connects to, as explained in class).

Here, we now have a common attribute (column) between the VENDOR table and the CUSTOMER table. This way we can perform joins on these two tables (as explained in class).

9. Next, describe the customer table to understand how a foreign key is reflected (MUL).

```
mysql> DESC CUSTOMER;
+-----+-----+-----+-----+-----+-----+
| Field      | Type        | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| CUSTOMER_ID | int         | NO   | PRI | NULL    |       |
| CUSTOMER_NAME | varchar(35) | YES  |     | NULL    |       |
| VENDOR_NO    | int         | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

B. JOINING TABLES

In this section we'll see how to join two tables, i.e., select data from two separate tables.

First, let's view the data in our student, course and registration tables. As we execute the join queries, look at the data in the tables being joined to understand what is happening.

```
mysql> SELECT * FROM course;
+-----+-----+-----+-----+
| course_code | course_name          | course_diagonal | course_coordinator |
+-----+-----+-----+-----+
| COMS2002    | Database Fundamentals | B               | Olaperi            |
| COMS2013    | Mobile Computing     | B               | Pravesh            |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

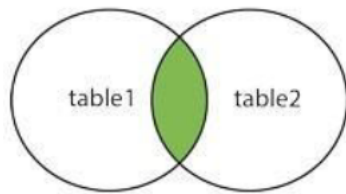
mysql> SELECT * FROM registration;
+-----+-----+-----+-----+
| course_code | student_no | registration_no | registration_date |
+-----+-----+-----+-----+
| COMS2002    | 11000010   | 111             | 2018-04-09        |
| COMS2013    | 11000010   | 112             | 2018-04-09        |
| COMS2002    | 11000011   | 113             | 2018-04-10        |
| COMS2013    | 11000011   | 114             | 2018-04-11        |
| COMS2002    | 11000012   | 115             | 2018-04-11        |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM student;
+-----+-----+-----+-----+
| student_no | student_fname | student_lname | student_contact |
+-----+-----+-----+-----+
| 11000010   | Henry         | Peter         | 0767263611      |
| 11000011   | Joe           | Sue           | 0738781982      |
| 11000012   | Jake          | Henry         | 0767136971      |
| 11000013   | Thabo         | Mbatha        | 0638220136      |
| 11000014   | Naledi        | Dlamini       | 0726268245      |
| 11000015   | Senzo         | Mokoena       | 0820436231      |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

As discussed in class, there are different types of join. In the SQL parlance, they are referred to as INNER JOIN (or JOIN), LEFT OUTER JOIN, RIGHT OUTER JOIN, AND FULL OUTER JOIN (FULL JOIN).

10. First, let's use the INNER JOIN.

INNER JOIN



11. Let's display student and registration details for all students who have registered.

In the syntax, you select from the first table, give the join clause (INNER JOIN) and then specify the second table. The ON command is used to indicate the common columns on which the join should be performed. You specify the tablename, a period and then the column name (e.g. student.student_no).

```
SELECT * FROM student
INNER JOIN registration
ON student.student_no = registration.student_no;
```

```
mysql> SELECT * FROM student
N regis -> JOIN registration
-> WHERE student.student_no = registration.student_no;
```

student_no	student_fname	student_lname	student_contact	course_code	student_no	registration_no	registration_date
11000010	Henry	Peter	0767263611	COMS2002	11000010	111	2018-04-09
11000010	Henry	Peter	0767263611	COMS2013	11000010	112	2018-04-09
11000011	Joe	Sue	0738781982	COMS2002	11000011	113	2018-04-10
11000011	Joe	Sue	0738781982	COMS2013	11000011	114	2018-04-11
11000012	Jake	Henry	0767136971	COMS2002	11000012	115	2018-04-11

5 rows in set (0.00 sec)

Notice that only the rows where the **student_no** appears in both tables are displayed (as expected of the inner join). Students with numbers 11000013, 11000014, and 11000015 are not shown because they only occur in one table and not in the other table. The inner join is like an **intersection** of two tables.

12. You can also specify only the columns you wish to view. Let's select only the **student.student_no**, **student_fname** and **registration date**. (Recall that these table are on different tables, hence the need to use JOIN). Since the column **student_no** exists in both tables, you need to qualify the column by adding the table name followed by a period, that way, the SQL engine knows which column you are referring to.

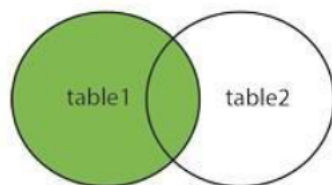
```
SELECT student.student_no, student_fname,
registration_date FROM student
INNER JOIN registration
ON student.student_no = registration.student_no;
```

student_no	student_fname	registration_date
11000010	Henry	2018-04-09
11000010	Henry	2018-04-09
11000011	Joe	2018-04-10
11000011	Joe	2018-04-11
11000012	Jake	2018-04-11

5 rows in set (0.00 sec)

13. Next, let's see the LEFT OUTER JOIN

Left Outer JOIN



14. Display all student and registration details for all students who have registered, including the students who are yet to register.

```
SELECT * FROM student
LEFT OUTER JOIN registration
ON student.student_no = registration.student_no;
```

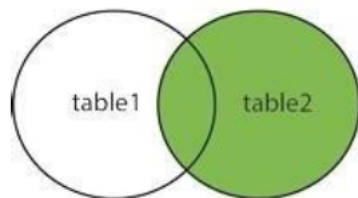
```
mysql> SELECT * FROM student
-> LEFT OUTER JOIN registration
-> ON student.student_no = registration.student_no;
```

student_no	student_fname	student_lname	student_contact	course_code	student_no	registration_no	registration_date
11000010	Henry	Peter	0767263611	COMS2002	11000010	111	2018-04-09
11000010	Henry	Peter	0767263611	COMS2013	11000010	112	2018-04-09
11000011	Joe	Sue	0738781982	COMS2002	11000011	113	2018-04-10
11000011	Joe	Sue	0738781982	COMS2013	11000011	114	2018-04-11
11000012	Jake	Henry	0767136971	COMS2002	11000012	115	2018-04-11
11000013	Thabo	Mbatha	0638220136	NULL	NULL	NULL	NULL
11000014	Naledi	Dlamini	0726268245	NULL	NULL	NULL	NULL
11000015	Senzo	Mokoena	0820436231	NULL	NULL	NULL	NULL

As you can see from the results, it shows the details of all students in the students table and their matching records in the registration table where such exists. Where a student in the student table does not have a record in the registration table, the columns from the registration table are left as NULL.

15. Next, let's see the RIGHT OUTER JOIN. A right outer join performs similarly to the left outer join, however, it will return all the rows from the second table in the query (registration in this case) rather than all the rows from the first table in the query (student) in this case.

Right Outer JOIN



```
SELECT * FROM student
RIGHT OUTER JOIN registration
ON student.student_no = registration.student_no;
```

student_no	student_fname	student_lname	student_contact	course_code	student_no	registration_no	registration_date
11000010	Henry	Peter	0767263611	COMS2002	11000010	111	2018-04-09
11000010	Henry	Peter	0767263611	COMS2013	11000010	112	2018-04-09
11000011	Joe	Sue	0738781982	COMS2002	11000011	113	2018-04-10
11000011	Joe	Sue	0738781982	COMS2013	11000011	114	2018-04-11
11000012	Jake	Henry	0767136971	COMS2002	11000012	115	2018-04-11

5 rows in set (0.00 sec)

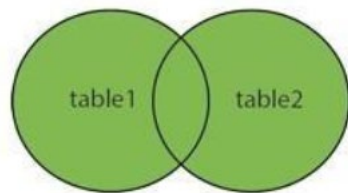
Of course, because there is no student in the registration table that is not in the student table, we do not see any NULL/empty values.

16. Next let's look at the FULL OUTER JOIN. The full outer join is used to select all records from both tables. In MySQL there is no FULL JOIN syntax. However, a full join is a union of both a left join and a right join, so let's see how to perform the full join in MySQL.

Let's perform a full outer join between the student and registration tables.

```
SELECT * FROM student
LEFT OUTER JOIN registration
ON student.student_no = registration.student_no
UNION
SELECT * FROM student
RIGHT OUTER JOIN registration
ON student.student_no = registration.student_no;
```

Full Outer JOIN



student_no	student_fname	student_lname	student_contact	course_code	student_no	registration_no	registration_date
11000010	Henry	Peter	0767263611	COMS2013	11000010	112	2018-04-09
11000010	Henry	Peter	0767263611	COMS2002	11000010	111	2018-04-09
11000011	Joe	Sue	0738781982	COMS2013	11000011	114	2018-04-11
11000011	Joe	Sue	0738781982	COMS2002	11000011	113	2018-04-10
11000012	Jake	Henry	0767136971	COMS2002	11000012	115	2018-04-11
11000013	Thabo	Mbatha	0638220136	NULL	NULL	NULL	NULL
11000014	Naledi	Dlamini	0726268245	NULL	NULL	NULL	NULL
11000015	Senzo	Mokoena	0820436231	NULL	NULL	NULL	NULL

If there were additional rows in the registration table not in the student table, those rows would also have been included.

PART 2 – YOUR TASK

INSTRUCTIONS

1. Write the SQL commands to perform the following tasks. Upload all the files into the Gradescope assignment – **Lab 5**.

QUESTIONS

1. Note that we'll be using the **student**, **registration** and **course** tables as in Lab 4.
2. Write the SQL command to modify the **student** table. Specify the **student_no** attribute as the primary key. Save your command in **lab5_q1.txt**

Expected Output when you describe the table:

```
mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| student_no     | char(8)       | NO   | PRI | NULL    |       |
| student_fname  | varchar(25)   | YES  |     | NULL    |       |
| student_lname  | varchar(25)   | YES  |     | NULL    |       |
| student_contact | char(11)      | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

3. Write the SQL command to modify the **course** table. Specify the **course_code** attribute as the primary key. Save your command in **lab5_q2.txt**

Expected Output when you describe the table:

```
mysql> desc course;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| course_code    | char(8)       | NO   | PRI | NULL    |       |
| course_name    | varchar(50)   | YES  |     | NULL    |       |
| course_diagonal | char(1)       | YES  |     | NULL    |       |
| course_coordinator | varchar(15)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

4. Write the SQL command to modify the **registration** table. Specify the **registration_no** attribute as the primary key. Specify the **course_code** from the **course** table and the **student_no** attribute from the **student** table as foreign keys. All these must be done in a single SQL statement, i.e., you must have only one semi-colon ';'. Save your command in **lab5_q3.txt**

Expected Output when you describe the table:

```
mysql> desc registration;
```

Field	Type	Null	Key	Default	Extra
course_code	char(8)	YES	MUL	NULL	
student_no	char(8)	YES	MUL	NULL	
registration_no	char(3)	NO	PRI	NULL	
registration_date	date	YES		NULL	

4 rows in set (0.00 sec)

5. Insert the following data into the **course** table.

course_code: COMS2004

course_name: Introduction to Computing

course_diagonal: B

course_coordinator: Refilwe

Save your command in **lab5_q4.txt**

Data in your course table should now be similar to this:

course_code	course_name	course_diagonal	course_coordinator
COMS2002	Database Fundamentals	B	Olaperi
COMS2004	Introduction to Computing	B	Refilwe
COMS2013	Mobile Computing	B	Pravesh

3 rows in set (0.00 sec)

6. Write the SQL command to display all courses (course table) for which students have registered (registration table). Save your command in **lab5_q5.txt**

Expected Output

course_code	course_name	course_diagonal	course_coordinator	course_code	student_no	registration_no	registration_date
COMS2002	Database Fundamentals	B	Olaperi	COMS2002	11000010	111	2018-04-09
COMS2013	Mobile Computing	B	Pravesh	COMS2013	11000010	112	2018-04-09
COMS2002	Database Fundamentals	B	Olaperi	COMS2002	11000011	113	2018-04-10
COMS2013	Mobile Computing	B	Pravesh	COMS2013	11000011	114	2018-04-11
COMS2002	Database Fundamentals	B	Olaperi	COMS2002	11000012	115	2018-04-11

5 rows in set (0.00 sec)

7. Write the SQL command to show all courses alongside students' registration details for those courses where such exists. Save your command in **lab5_q6.txt**

Expected Output

course_code	course_name	course_diagonal	course_coordinator	course_code	student_no	registration_no	registration_date
COMS2002	Database Fundamentals	B	Olaperi	COMS2002	11000012	115	2018-04-11
COMS2002	Database Fundamentals	B	Olaperi	COMS2002	11000011	113	2018-04-10
COMS2002	Database Fundamentals	B	Olaperi	COMS2002	11000010	111	2018-04-09
COMS2004	Introduction to Computing	B	Refilwe	NULL	NULL	NULL	NULL
COMS2013	Mobile Computing	B	Pravesh	COMS2013	11000011	114	2018-04-11
COMS2013	Mobile Computing	B	Pravesh	COMS2013	11000010	112	2018-04-09

6 rows in set (0.00 sec)

8. Write the SQL command to perform a full outer join between the registration and course tables. Save your command in **lab5_q7.txt**

Expected output

course_code	student_no	registration_no	registration_date	course_code	course_name	course_diagonal	course_coordinator
COMS2002	11000010	111	2018-04-09	COMS2002	Database Fundamentals	B	Olaperi
COMS2013	11000010	112	2018-04-09	COMS2013	Mobile Computing	B	Pravesh
COMS2002	11000011	113	2018-04-10	COMS2002	Database Fundamentals	B	Olaperi
COMS2013	11000011	114	2018-04-11	COMS2013	Mobile Computing	B	Pravesh
COMS2002	11000012	115	2018-04-11	COMS2002	Database Fundamentals	B	Olaperi
NULL	NULL	NULL	NULL	COMS2004	Introduction to Computing	B	Refilwe

6 rows in set (0.00 sec)

Reference on Foreign keys

<https://dev.mysql.com/doc/refman/8.0/en/create-table-foreign-keys.html>

Reference on Joins

<https://dev.mysql.com/doc/refman/8.0/en/join.html>