

| DATA ANALYTICS REFERENCE DOCUMENT |                                    |
|-----------------------------------|------------------------------------|
| <b>Document Title:</b>            | Applied Databases week 5 worksheet |
| <b>Document No.:</b>              | 1550745911                         |
| <b>Author(s):</b>                 | Gerhard van der Linde              |
| <b>Contributor(s):</b>            |                                    |

#### REVISION HISTORY

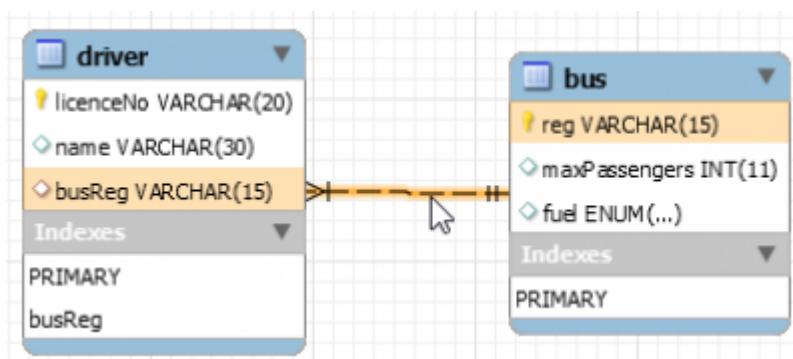
| Revision | Details of Modification(s) | Reason for modification            | Date             | By                    |
|----------|----------------------------|------------------------------------|------------------|-----------------------|
| 0        | Draft release              | Applied Databases week 5 worksheet | 2019/02/21 10:45 | Gerhard van der Linde |

## Applied Databases - Week 5

### 1. Get bus.sql from Moodle and import it into MySQL.

```
MySQL -u root -p <bus.sql
```

### 2. How are the tables in the database related?



```
mysql> show create table driver;
+-----+-----+
| Table | Create Table |
+-----+-----+
| driver | CREATE TABLE `driver` (
  `licenceNo` varchar(20) NOT NULL,
  `name` varchar(30) DEFAULT NULL,
  `busReg` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`licenceNo`),
  KEY `busReg` (`busReg`),
  CONSTRAINT `driver_ibfk_1`
  FOREIGN KEY (`busReg`) REFERENCES `bus` (`reg`) ON DELETE CASCADE
)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+-----+
1 row in set (0.00 sec)
```

The driver has a foreign key constraint busReg that requires a valid entry that already exists in the table bus.reg to be present. When the bus entry is deleted from thus bus table, the “ON DELET CASCADE” will also delete the driver associated with this bus entry.

### 3. Add the following drivers:

- “Mary”
- “Bob” – licence number “RN2423”
- “Sean” – licence number “FF88345” who drives bus “191-G-123”
- What happens and why?

```
mysql> insert into driver (name) values("Mary");
ERROR 1364 (HY000): Field 'licenceNo' doesn't have a default value
```

This insert fails because the licenseNo field is a primary key value and cannot be NULL and fails when no value is assigned during the insert.

```
mysql> insert into driver (licenceNo,name) values("RN2423","Bob");
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from driver;
+-----+-----+-----+
| licenceNo | name | busReg |
+-----+-----+-----+
| F2233     | Alan | 191-G-123 |
| L23423    | John | 12-G-1323 |
| RN2423    | Bob  | NULL      |
| X98983    | Tom  | 161-D-1323 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

The second query works since Bo is added with a licensee number and the minimum criteria is being met.

```
mysql> insert into driver (licenceNo,name,busReg) values("FF88345","Sean","191-G-123");
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from driver;
+-----+-----+-----+
| licenceNo | name | busReg |
+-----+-----+-----+
| F2233     | Alan | 191-G-123 |
| FF88345    | Sean | 191-G-123 |
| L23423    | John | 12-G-1323 |
| RN2423    | Bob  | NULL      |
| X98983    | Tom  | 161-D-1323 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

The last insert also works since all the fields are populated with valid values.

## 4. Add the following buses:

- "12-G-1323" that can hold up to 34 passengers and runs on "Diesel"
- "171-G-885" that can hold up to 84 passengers and runs on "Petrol"
- "191-D-45890" that can hold up to 120 passengers and runs on "Ethanol"
- What happens and why?

```
mysql> select * from bus;
+-----+-----+-----+
| reg      | maxPassengers | fuel   |
+-----+-----+-----+
| 12-G-1323 | 34            | Petrol |
| 161-D-1323 | 80            | Diesel |
| 162-D-3433 | 120           | Electric |
| 191-G-123  | 56            | Diesel |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> insert into bus values("12-G-1323",34,"Diesel");
ERROR 1062 (23000): Duplicate entry '12-G-1323' for key 'PRIMARY'
```

The bus already exists in the table and the primary key constraint prevents duplication in this reg column.

```
mysql> insert into bus values("171-G885",84,"Petrol");
Query OK, 1 row affected (0.01 sec)

mysql> select * from bus;
+-----+-----+-----+
| reg      | maxPassengers | fuel   |
+-----+-----+-----+
| 12-G-1323 | 34            | Petrol |
| 161-D-1323 | 80            | Diesel |
| 162-D-3433 | 120           | Electric |
| 171-G885  | 84            | Petrol |
| 191-G-123  | 56            | Diesel |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

All the criteria entered meets the requirements and is successfully added to the table.

```
mysql> insert into bus values("191-D-45890",120,"Ethanol");
ERROR 1265 (01000): Data truncated for column 'fuel' at row 1
mysql> select * from bus;
+-----+-----+-----+
| reg      | maxPassengers | fuel   |
+-----+-----+-----+
| 12-G-1323 | 34            | Petrol |
| 161-D-1323 | 80            | Diesel |
| 162-D-3433 | 120           | Electric |
| 171-G885  | 84            | Petrol |
| 191-G-123  | 56            | Diesel |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Ethanol is not in the enumeration list of fuels and is therefore rejected.

1. `CREATE TABLE `bus` (`
2. ``reg` varchar(15) NOT NULL,`
3. ``maxPassengers` int(11) DEFAULT NULL,`
4. ``fuel` enum('Diesel','Petrol','Electric') DEFAULT 'Diesel',`

5. PRIMARY KEY (`reg`)
6. ) ENGINE=InnoDB DEFAULT CHARSET=latin1 |

## 5. Update driver's licences that contain the letters "F" or "R" to have the letters "T-" before their current licence number.

```
Update driver
set licenceNo = concat("T-", licenceNo)
where licenceNo like "F%"
or licenceNo like "R%";
```

```
select * from driver;
```

| licenceNo | name | busReg     |
|-----------|------|------------|
| L23423    | John | 12-G-1323  |
| T-F2233   | Alan | 191-G-123  |
| T-FF88345 | Sean | 191-G-123  |
| T-RN2423  | Bob  | NULL       |
| X98983    | Tom  | 161-D-1323 |

```
5 rows in set (0.00 sec)
```

## 6. Delete driver "Alan".

What happens and why?

```
mysql> select * from driver;
```

| licenceNo | name | busReg     |
|-----------|------|------------|
| L23423    | John | 12-G-1323  |
| T-F2233   | Alan | 191-G-123  |
| T-FF88345 | Sean | 191-G-123  |
| T-RN2423  | Bob  | NULL       |
| X98983    | Tom  | 161-D-1323 |

```
5 rows in set (0.00 sec)
```

```
mysql> select * from bus;
```

| reg        | maxPassengers | fuel     |
|------------|---------------|----------|
| 12-G-1323  | 34            | Petrol   |
| 161-D-1323 | 80            | Diesel   |
| 162-D-3433 | 120           | Electric |
| 171-G885   | 84            | Petrol   |
| 191-G-123  | 56            | Diesel   |

```
5 rows in set (0.00 sec)
```

```
mysql> delete from driver
```

```
-> where name = "Alan";
```

Query OK, 1 row affected (0.01 sec)

```
mysql> select * from driver;
```

| licenceNo | name | busReg     |
|-----------|------|------------|
| L23423    | John | 12-G-1323  |
| T-FF88345 | Sean | 191-G-123  |
| T-RN2423  | Bob  | NULL       |
| X98983    | Tom  | 161-D-1323 |

4 rows in set (0.00 sec)

```
mysql> select * from bus;
```

| reg        | maxPassengers | fuel     |
|------------|---------------|----------|
| 12-G-1323  | 34            | Petrol   |
| 161-D-1323 | 80            | Diesel   |
| 162-D-3433 | 120           | Electric |
| 171-G885   | 84            | Petrol   |
| 191-G-123  | 56            | Diesel   |

5 rows in set (0.00 sec)

Alan was deleted and nothing else happens. Assuming the question is asked to highlight the working of the foreign key constraint busReg and the "ON DELETE CASCADE" clause.



So in this case nothing happens but if the bus assigned to Alan was deleted Alan would have disappeared too. Is this right?

## 7. Delete bus "161-d-1323".

What happens and why?

```
mysql> select * from bus;
```

| reg        | maxPassengers | fuel     |
|------------|---------------|----------|
| 12-G-1323  | 34            | Petrol   |
| 161-D-1323 | 80            | Diesel   |
| 162-D-3433 | 120           | Electric |
| 171-G885   | 84            | Petrol   |
| 191-G-123  | 56            | Diesel   |

5 rows in set (0.00 sec)

```
mysql> select * from driver;
```

| licenceNo | name | busReg     |
|-----------|------|------------|
| L23423    | John | 12-G-1323  |
| T-FF88345 | Sean | 191-G-123  |
| T-RN2423  | Bob  | NULL       |
| X98983    | Tom  | 161-D-1323 |

```
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> delete from bus
-> where reg = "161-D-1323";
Query OK, 1 row affected (0.01 sec)

mysql> select * from bus;
+-----+-----+-----+
| reg          | maxPassengers | fuel      |
+-----+-----+-----+
| 12-G-1323    | 34            | Petrol    |
| 162-D-3433   | 120           | Electric  |
| 171-G885     | 84            | Petrol    |
| 191-G-123    | 56            | Diesel    |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from driver;
+-----+-----+-----+
| licenceNo | name  | busReg    |
+-----+-----+-----+
| L23423    | John  | 12-G-1323 |
| T-FF88345 | Sean  | 191-G-123 |
| T-RN2423  | Bob   | NULL      |
+-----+-----+-----+
3 rows in set (0.00 sec)
```



So bus "161-D-1323" and driver "Tom" was deleted because of the "ON DELETE CASCADE" foreign key reference applied on the driver table.

## 8. Get bus2.sql from Moodle and import it into MySQL.

To use this database type use bus2;

```
MySQL -u root -p <bus2.mysql
```

## 9. Delete bus "161-d-1323".

What happens and why?

```
mysql> use bus2;
Database changed
mysql> select * from driver;
+-----+-----+-----+
| licenceNo | name  | busReg    |
+-----+-----+-----+
| F2233     | Alan  | 191-G-123 |
| L23423    | John  | 12-G-1323 |
| X98983    | Tom   | 161-D-1323 |
+-----+-----+-----+
```

```

3 rows in set (0.00 sec)

mysql> select * from bus;
+-----+-----+-----+
| reg      | maxPassengers | fuel    |
+-----+-----+-----+
| 12-G-1323 | 34            | Petrol  |
| 161-D-1323 | 80            | Diesel  |
| 162-D-3433 | 120           | Electric |
| 191-G-123  | 56            | Diesel  |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> delete from bus
-> where reg = "161-D-1323";
Query OK, 1 row affected (0.01 sec)

mysql> select * from driver;
+-----+-----+-----+
| licenceNo | name  | busReg |
+-----+-----+-----+
| F2233     | Alan  | 191-G-123 |
| L23423     | John  | 12-G-1323 |
| X98983     | Tom   | NULL     |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from bus;
+-----+-----+-----+
| reg      | maxPassengers | fuel    |
+-----+-----+-----+
| 12-G-1323 | 34            | Petrol  |
| 162-D-3433 | 120           | Electric |
| 191-G-123  | 56            | Diesel  |
+-----+-----+-----+
3 rows in set (0.00 sec)

```



The behavior changed by setting driver Tom busReg foreign key to NULL. This happened because of the amended foreign key clause "ON DELETE SET NULL"

```

mysql> show create table driver;
CREATE TABLE `driver` (
  `licenceNo` varchar(20) NOT NULL,
  `name` varchar(30) DEFAULT NULL,
  `busReg` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`licenceNo`),
  KEY `busReg` (`busReg`),
  CONSTRAINT `driver_ibfk_1` FOREIGN KEY (`busReg`)
  REFERENCES `bus` (`reg`) ON DELETE SET NULL
)

```

## 10. Get employees2.sql from Moodle and import it into MySQL.

To use this database type use employees2;

```
MySQL -u root -p <employees2.sql
use employees2;
Database changed
```

## 11. Show the emp\_no, first\_name and last\_name of employees born in the average year.

The average year should be rounded down to the nearest whole number.

For example,

- 1949.1 becomes 1949.
- 1949.9 becomes 1949.
- 1949.0 becomes 1949.

```
mysql> SELECT emp_no, first_name, last_name, year(birth_date) as bd
-> FROM employees2.employees
-> where year(birth_date) in (
->   SELECT truncate(avg(year(birth_date)),0) FROM employees2.employees
-> )
-> ;
```

| emp_no | first_name | last_name    | bd   |
|--------|------------|--------------|------|
| 10007  | Tzvetan    | Zielinski    | 1957 |
| 10045  | Moss       | Shanbhogue   | 1957 |
| 10054  | Mayumi     | Schuessler   | 1957 |
| 10080  | Premal     | Baek         | 1957 |
| 10094  | Arumugam   | Ossenbruggen | 1957 |

5 rows in set (0.00 sec)

or with rounding instead ....

```
mysql> SELECT emp_no, first_name, last_name, year(birth_date) as bd
-> FROM employees2.employees
-> where year(birth_date) in (
->   SELECT round(avg(year(birth_date)),0) FROM employees2.employees
-> )
-> ;
```

| emp_no | first_name | last_name | bd   |
|--------|------------|-----------|------|
| 10008  | Saniya     | Kalloufi  | 1958 |
| 10017  | Cristinel  | Bouloucos | 1958 |
| 10024  | Suzette    | Pettey    | 1958 |
| 10025  | Prasadram  | Heyers    | 1958 |
| 10030  | Elvis      | Demeyer   | 1958 |
| 10050  | Yinghua    | Dredge    | 1958 |
| 10071  | Hisao      | Lipner    | 1958 |

7 rows in set (0.00 sec)



## 12. Show the emp\_no, first\_name, last\_name and name of the department each employee is in.

First determine the relationships....

```
mysql> show tables;
```

```
+-----+
| Tables_in_employees2 |
+-----+
| dept                  |
| employees             |
| salaries              |
+-----+
3 rows in set (0.00 sec)
```

```
mysql> show create table dept;
```

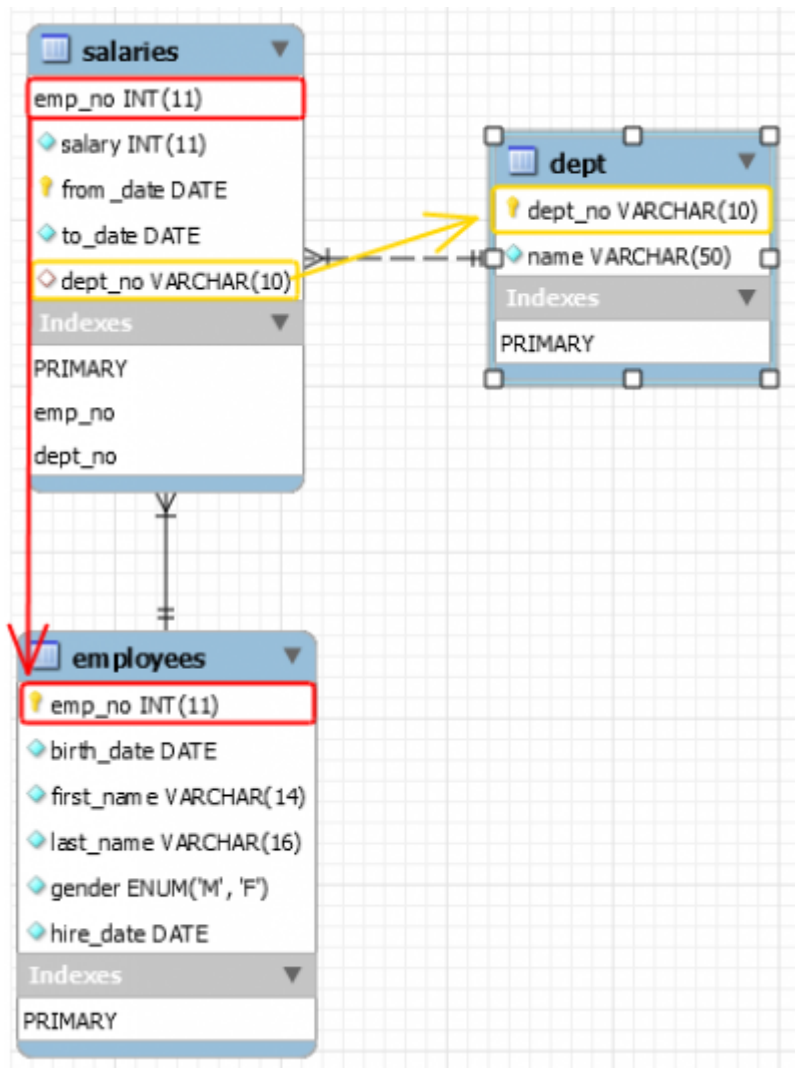
```
CREATE TABLE `dept` (
  `dept_no` varchar(10) NOT NULL,
  `name` varchar(50) NOT NULL,
  PRIMARY KEY (`dept_no`)
)
```

```
mysql> show create table employees;
```

```
CREATE TABLE `employees` (
  `emp_no` int(11) NOT NULL,
  `birth_date` date NOT NULL,
  `first_name` varchar(14) NOT NULL,
  `last_name` varchar(16) NOT NULL,
  `gender` enum('M','F') NOT NULL,
  `hire_date` date NOT NULL,
  PRIMARY KEY (`emp_no`)
)
1 row in set (0.00 sec)
```

```
mysql> show create table salaries;
```

```
CREATE TABLE `salaries` (
  `emp_no` int(11) NOT NULL,
  `salary` int(11) NOT NULL,
  `from_date` date NOT NULL,
  `to_date` date NOT NULL,
  `dept_no` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`emp_no`,`from_date`),
  KEY `emp_no` (`emp_no`),
  KEY `dept_no` (`dept_no`),
  CONSTRAINT `salaries_ibfk_1` FOREIGN KEY (`emp_no`) REFERENCES `employees` (`emp_no`) ON
DELETE CASCADE,
  CONSTRAINT `salaries_ibfk_2` FOREIGN KEY (`dept_no`) REFERENCES `dept` (`dept_no`) ON DELETE
CASCADE
)
1 row in set (0.00 sec)
```



So the master table is salaries and contains foreign key to the employees and departments tables.

## Construct the query

```
SELECT DISTINCT(ee.emp_no), ee.first_name, ee.last_name, ed.name AS department
FROM employees2.employees AS ee
LEFT JOIN employees2.salaries AS es
ON ee.emp_no=es.emp_no
LEFT JOIN employees2.dept AS ed
ON es.dept_no=ed.dept_no
```

## Concatenated Query Result

| emp_no | first_name | last_name | department      |
|--------|------------|-----------|-----------------|
| 10001  | Georgi     | Facello   | Human Resources |
| 10002  | Bezalel    | Simmel    | Human Resources |
| 10003  | Parto      | Bamford   | Human Resources |
| ...    |            |           |                 |
| 10028  | Domenick   | Tempesti  | Human Resources |

| emp_no | first_name  | last_name  | department             |
|--------|-------------|------------|------------------------|
| 10029  | Otmar       | Herbst     | Human Resources        |
| 10030  | Elvis       | Demeyer    | Human Resources        |
| 10031  | Karsten     | Joslin     | Research & Development |
| 10032  | Jeong       | Reistad    | Research & Development |
| 10033  | Arif        | Merlo      | Research & Development |
| ...    |             |            |                        |
| 10068  | Charlene    | Brattka    | Research & Development |
| 10069  | Margareta   | Bierman    | Research & Development |
| 10070  | Reuven      | Garigliano | Research & Development |
| 10071  | Hisao       | Lipner     | Sales                  |
| 10072  | Hironoby    | Sidou      | Sales                  |
| 10073  | Shir        | McClurg    | Sales                  |
| ...    |             |            |                        |
| 10098  | Sreekrishna | Servieres  | Sales                  |
| 10099  | Valter      | Sullins    | Sales                  |
| 10100  | Hironobu    | Haraldson  | Sales                  |

100 rows in set (0.00 sec)

From:

<http://hdip-data-analytics.com/> - HDip Data Analytics

Permanent link:

<http://hdip-data-analytics.com/submissions/worksheet/databases/topic5>

Last update: **2020/06/20 14:39**