Module 11 Relational Databases

Module title	Relational Databases
Module NFQ level (only if an NFQ level can be demonstrated)	6
Module number/reference	BSCH-RD
Parent programme(s)	Bachelor of Science (Honours) in Computing Science
Stage of parent programme	Stage 2
Semester (semester1/semester2 if applicable)	Semester 1
Module credit units (FET/HET/ECTS)	ECTS
Module credit number of units	10
List the teaching and learning modes	Direct, Blended
Entry requirements (statement of knowledge, skill and competence)	Learners must have achieved programme entry requirements
Pre-requisite module titles	None
Co-requisite module titles	None
Is this a capstone module? (Yes or No)	No
Specification of the qualifications (academic, pedagogical and professional/occupational) and experience required of staff (staff includes workplace personnel who are responsible for learners such as apprentices, trainees and learners in clinical placements)	Qualified to as least a Bachelor of Science (Honours) level in Computer Science or equivalent and with a Certificate in Training and Education (30 ECTS at level 9 on the NFQ) or equivalent.
Maximum number of learners per centre (or instance of the module)	60
Duration of the module	Two Academic Semesters, 12 weeks teaching
Average (over the duration of the module) of the contact hours per week	4.5
Module-specific physical resources and support required per centre (or instance of the module)	One class room with capacity for 60 learners along with one computer lab with capacity for 25 learners for each group of 25 learners

Analysis of required learning effort		
	Minimum ratio teacher / learner	**Hours**
Effort while in contact with staff		
Classroom and demonstrations	1:60	18
Monitoring and small-group teaching	1:25	36
Other (specify)		
Independent Learning		
Directed e-learning		
Independent Learning		100
Other hours (worksheets and assignments)		96
Work-based learning – learning effort		
Total Effort		250

Allocation of marks (within the module)					
	Continuous assessment	**Supervised project**	**Proctored practical examination**	**Proctored written examination**	**Total**
Percentage contribution	50%			50%	100%

Module aims and objectives

The aim of this module is to teach the theoretical and practical underpinnings of modern database management systems by teaching the design and implementation of Relational Databases. Learners study techniques such as entity-relationship modelling and normalisation in order to more effectively design a database. Learners are also exposed to practical application of structured query language (SQL) in order to implement, populate, query, and manipulate a database design into a relational database management system (RDBMS).

The objectives of the module are to give learners the skills to model databases using E-R diagrams and ensure database designs do not have redundancy through the process of normalisation. Through the practical application of SQL, learners are able to create, update, retrieve, and delete data from databases that implement their database designs. We also enable learners to integrate their databases with other components (self-developed programs) as part of the software development process.

Minimum intended module learning outcomes
On successful completion of this module, the learner will be able to:

1. Discuss the goals, functions, models, and components of database systems
2. Explain the context, phases, and techniques for designing and building databases
3. Select and apply appropriate design models to a given development environment
4. Design an efficient database system for a business functional area using an appropriate database design technique
5. Implement and manipulate a relational database using SQL
6. Discuss advanced topics in database systems

Rationale for inclusion of the module in the programme and its contribution to the overall MIPLOs
The module enables learners to store and manipulate data in a format that is used as the basis of dynamic websites, data analytics, and data science that does not involve a set of files in a filesystem. Appendix 1 of the programme document maps MIPLOs to the modules through which they are delivered.

Information provided to learners about the module
Learners receive a programme handbook to include module descriptor, module learning outcomes (MIMLO), class plan, assignment briefs, assessment strategy and reading materials.

Module content, organisation and structure
Database Management System (DBMS)

- Generalised functions of a DBMS
- Components of a DBMS
- File organisation within a DBMS

The Relational Model

- Relational Algebra
- ER Modelling, diagrams and mapping to a relational schema,
- Data normalisation,
- Relational integrity, keys, indexes,

SQL

- Schema definition and data manipulation in SQL,
- SQL queries,
- Aggregates and groups

- Joins and nested queries
- An introduction to stored procedures

Non-relational storage

- schema-less storage (NoSQL);
- XML and XML Schema
- Object Oriented databases

Module teaching and learning (including formative assessment) strategy

The module is taught as a combination of lectures and lab sessions. The lecture sessions discuss and explain to learners the theoretical underpinnings of how a Relational Database and RDBMS works and functions along with best practices for designing databases for these systems. The practical lab sessions give learners an opportunity to interact with an RDBMS and learn the skills of creating, updating, retrieving, and deleting data from a database, as well as designing databases from scratch and normalising databases.

Assessment is divided into three. First there are a number of weekly worksheets that will build the learners skills with a relational database, E-R diagrams and normalisation. There is a major assignment after this through which learners will be provided with a sample database design problem and will be expected to execute all steps to produce a fully working, normalised database filled with data and working queries. Finally, there is an end of semester exam that tests the learners understanding of the theoretical material.

Timetabling, learner effort and credit

The module is timetabled as one 1.5-hour lecture and two 1.5-hour labs per week.

The number of 10 ECTS credits assigned to this module is our assessment of the amount of learner effort required. Continuous assessment spreads the learner effort to focus on small steps before integrating all steps into the overall process of database design and implementation.

There are 48 contact hours made up of 12 lectures delivered over 12 weeks with classes taking place in a classroom. There are also 24 lab sessions delivered over 12 weeks taking place in a fully equipped computer lab. The learner will need 100 hours of independent effort to further develop the skills and knowledge gained through the contact hours. An additional 96 hours are set aside for learners to work on worksheets and assignments that must be completed for the module.

The team believes that 250 hours of learner effort are required by learners to achieve the MIMLOs and justify the award of 10 ECTS credits at this stage of the programme.

Work-based learning and practice-placement

There is no work based learning or practice placement involved in the module.

E-learning

The college VLE is used to disseminate notes, advice, and online resources to support the learners. The learners are also given access to Lynda.com as a resource for reference.

Module physical resource requirements

Requirements are for a classroom for 60 learners equipped with a projector, and a 25-seater computer lab for practical sessions with access to SQLite and MySQL (this may change should better technologies arise).

Reading lists and other information resources

Recommended Text

Elmasri, R. and Navathe, S. B. (2016) *Fundamentals of Database Systems.* Boston: Pearson.

Jones, P. (2017) *Sql: The Fundamentals of SQL: A Complete Beginners Guide to SQL Mastery*. CreateSpace Independent Publishing Platform.

Secondary Reading:

Patrick, J. J. (2009) *SQL Fundamentals*. Upper Saddle River: Prentice Hall.

Date, C. J. (2004) *An Introduction to Database Systems*. Boston: Pearson Education.

Specifications for module staffing requirements

For each instance of the module, one lecturer qualified to at least Bachelor of Science (Honours) in Computer Science or equivalent, and with a Certificate in Training and Education (30 ECTS at level 9 on the NFQ) or equivalent.. Industry experience would be a benefit but is not a requirement.

Learners also benefit from the support of the programme director, programme administrator, learner representative and the Student Union and Counselling Service.

Module Assessment Strategy

The assignments constitute the overall grade achieved, and are based on each individual learner's work. The continuous assessments provide for ongoing feedback to the learner and relates to the module curriculum.

No.	Description	MIMLOs	Weighting
1	A series of weekly worksheets that will foster development of the skills of: • Database and table creation • Database population and querying • Database Design • Normalisation • SQL fundamentals, CRUD operations Queries involving table joins	1,2,3,5,6	35%
2	Major assignment that will assess if the learners can integrate the skills assessed in the worksheet into one large database design problem. Learners will be given a sample database design problem and will be required to design, normalise, initialise, populate and query the database	1-6	15%
3	Written exam that tests the theoretical aspects of the module	1-6	50%

All repeat work is capped at 40%.

Sample assessment materials

Note: All assignment briefs are subject to change in order to maintain current content.

RELATIONAL DATABASES PRACTICAL 1 WORKING WITH A SINGLE TABLE DATABASE.

Introduction:

In this practical you will generate a single table database containing a list of students and the courses that they are enrolled on. You will be required to become familiar with the commands for creating a table and inserting data into a table. You are also expected to use a set of default values for some of the fields (ask me how this works).

Notes:

You will be required to submit a pdf and your database inside an archive and submit to the moodle by Sunday 11th at 23:55. However, this should be easily doable within the two lab sessions that you have. You will be required to use sqlite to generate this database. Generate the database as shown below and answer the following questions about it

Task List:

01) create a table for students that contains the following information

- StudentID: primary key that is an integer and autoincrements
- FirstName: a string that is 10 characters in length
- LastName: a string that is 10 characters in length
- PhoneNumber: a string that is 15 characters in length
- Email: a string that is 30 characters in length
- CourseCode: a string that is 10 characters in length
- CourseName: a string that is 30 characters in length
- Duration: a number representing the length in weeks of a course. Should have a default value of 10
- CourseCost: a number repesenting the cost of a course. Should have a default value of 300.

in your PDF write out the create table command you used to perform this. (10%)

02) insert the following students into the database and write out the list of commands used to insert them. Points will be deducted if you fail to take advantage of default values. (20%)

Mary	Murphy	01-288 8888	Mmurphy@gcd.ie	ACC20	Access course	10	300
Brian	Smith	01-288 7777	Bsmith@gcd.ie	ACC20	Access course	10	300
David	Honan	01-288 6677	Dhonan@gcd.ie	WEB20	Web design	10	200
Frank	Murphy	01-233 5577	Fmurphy@gcd.ie	WEB20	Web design	10	200
Aoife	Byrne	01-233 5599	Abyrne@gcd.ie	WORD20	MS. Word	8	310
Cora	William	01-288 7723	Cwilliam@gcd.ie	ACC20	Access course	10	300
Conor	James	01-234 0023	Cjames@gcd.ie	EXL20	MS. Excel	9	320

03) try inserting David Honan into the database again with the same student ID. Show the command used to determine his student number and insert him into the database again. What happened and why? (20%)

04) this time try inserting David Honan into the database again but let the database assign him a student number. Show the command used. What happened? Why is this a bad state for the database? (10%)

05) What information is repeated in the table? How many pieces of information would have to be updated if the access course cost changed from 300 to 400? (20%)

06) If you were to delete Aoife Byrne from the database what information would go missing from the database with her that the college would like to keep?.(20%)

RELATIONAL DATABASES PRACTICAL 2

DECOMPOSING OUR SINGLE TABLE INTO TWO TABLES.

Introduction:

In this practical you will be required to take the single table that you generated for the last practical and decompose it into three smaller tables. The reason for this is that you had issues surrounding adding a student into more than one course at a time and also the loss of information about courses and costs with removing a student from the database. By decomposing our table into three smaller tables we will be able map students to multiple courses.

Notes:

You will be required to submit a pdf and your database inside an archive and submit to the moodle by Sunday 18th at 23:55. However, this should be easily doable within the two lab sessions that you have. You will be required to use sqlite to generate this database. Generate the database as shown below and answer the following questions about it

Task List:

01) create a table for students that contains the following information

- StudentID: primary key that is an integer and autoincrements
- FirstName: a string that is 10 characters in length
- LastName: a string that is 10 characters in length
- PhoneNumber: a string that is 15 characters in length
- Email: a string that is 30 characters in length

in your PDF write out the create table command you used to perform this (10%)

02) create a table for courses that contains the following information

- CourseCode: a string that is 10 characters in length
- CourseName: a string that is 30 characters in length
- Duration: a number representing the length in weeks of a course. Should have a default value of 10
- CourseCost: a number repesenting the cost of a course. Should have a default value of 300.

in your PDF write out the create table command you used to perform this. (10%)

03) create a third table that will act as a mapping between students and courses. In your PDF write out the create table command you used to perform this. (10%)

04) insert the following students into the database and write out the list of commands used to insert them. (10%)

Mary	Murphy	01-288 8888	Mmurphy@gcd.ie
Brian	Smith	01-288 7777	Bsmith@gcd.ie
David	Honan	01-288 6677	Dhonan@gcd.ie
Frank	Murphy	01-233 5577	Fmurphy@gcd.ie
Aoife	Byrne	01-233 5599	Abyrne@gcd.ie
Cora	William	01-288 7723	Cwilliam@gcd.ie
Conor	James	01-234 0023	Cjames@gcd.ie

05) insert the following information into the courses table and write out the list of commands used to insert them (10%)

ACC20	Access course	10	300
WEB20	Web design	10	200
WORD20	Microsoft Word	8	310
EXL20	Microsoft Excel	9	320

06) link up the following students to the following courses using the mapping table. Show the list of commands used to insert them. (10%)

- Mary Murphy (access course)
- Brian Smith (access course)
- David Honan (web design, access course)
- Frank Murphy (web design, access course)
- Aoife Byrne (word, excel)
- Cora William (access course)

- Conor James (excel)

07) Write queries (and show it in your PDF) that will (10%)

- pick out the names and email addresses of all students
- pick out the names of all courses, their duration and cost.

08) Write a query that will pick out the names of students who are taking the web design and access courses (you will be required to use the JOIN syntax here) (10%)

09) Is there any information repeated in the database? Explain the differences between the information that is repeated here compared to the information that was repeated in last weeks database? (10%)

10) Why do we need the foreign key and primary key restrictions on the mapping table? What would happen if we did not have these restrictions? (10%)

RELATIONAL DATABASES PRACTICAL 3: BUILDING A DATABASE WITH A THREE ATTRIBUTE MAPPING TABLE.

Introduction:

In this practical you will be generating a database for a car showroom that will involve four tables. In this database the car showroom needs to keep track of the cars it sells, the salesmen who sell the cars, and also the customers who buy the cars. Here you will see examples where we can answer questions about the data we have that do not involve all tables, thus reducing the amount of work that the database has to do to answer queries. Note that this time I will expect you to create the appropriate attributes and their types for tables yourself.

Notes:

You will be required to submit a pdf and your database inside an archive and submit to the moodle by Sunday 25th at 23:55. However, this should be easily doable within the two lab sessions that you have. You will be required to use sqlite to generate this database. Generate the database as shown below and answer the following questions about it

Task List:

01) create a table for salesmen with attributes for:, firstname, lastname, phonenumber, and starting date of employment. Show the create table and insert statements that you used to generate this table and populate it with the following information: (10%)

- John Doe, with a phonenumber of 555-123-4567 who started the first of january 2012
- Eric Bloggs, with a phone number of 555-123-4555 who started in feburary 2 years ago.
- Mark Smith, with a phone number of 555-123-4721 who started in june of 2010

02) create a table for cars with attributes for: make, model, date the model was first produced and price . Show the create table and insert statements that you used to generate this table and populate it with the following information: (10%)

- Mazda 6 (2nd generation) introduced in 2008 for 30,000
- Toyota Prius (1st generation) introduced in 1997 for 4,000
- Nissan GT-R (1st generation) introduced in 2007 for 100,000

03) create a table for customers with attributes for: firstname, lastname, phonenumber, address, email. Show the create table and insert statements that you used to generate this table and populate it with the following information. And populate it with the following information: (10%)

- Jerry Suave with a phone number of 555-412-3189 and email of js@bmail.com who lives in 1 Redmond Way.
- Brian Griffin with a phone number of 555-316-2179 and an email of bg@bmail.com who lives in 31 Spooner Street.
- Homer Simpson with a phone number of 555-814-5678 and an email of hs@bmail.com who lives in 742 Evergreen Terrace.
- Joe Swanson with a phone number of 555-316-2189 and an email of js@bmail.com who lives in 40 Spooner Street.

04) create a mapping table to link salesmen, cars, and customers together and give it attributes to store when a car was sold. Show the create table and insert statements that you used to generate this table and populate it with the following information: (10%)

- Mark Smith sold Jerry Suave a Nissan GTR in July of 2012
- Mark Smith sold Brian Griffin a Toyota Prius in 2013
- John Doe sold Homer Simpson a Mazda 6 in 2014
- Eric Bloggs sold Joe Swanson a Toyota Prius in 2015

05) write a query that will show how many cars each salesman sold and order them by the most number of sales first (ask me how to order attributes and results). Show your query and result (15%)

06) write a query that will show how many cars each customer bought from the show room order by highest number first. Show your query and result (15%)

07) write a query that will show the names of anyone who bought a Prius from this show room show your query and result. Order your results by firstname (15%)

08) write a query that will show the total value of car sales from each salesman and order them by highest value first. Show your query and result (15%)

RELATIONAL DATABASES PRACTICAL 4: INTRODUCTION TO ER DIAGRAMS AND CONVERSION TO DATABASES

Introduction:

In this practical you will be tasked with building a simple database from the spec that you will be given below. You are required to model this with an ER diagram (which must be included in your submission) and then convert this ER diagram into a database model. After this you will be expected to run some queries on your data and produce some useful information from it.

A company is starting out with using databases to store information. To begin with they want a simple database that will track their employees, departments and also the buildings they have. Employees will work in a department and there can be multiple departments in the same building. Departments are never split amongst multiple buildings. To begin with the following information must be inserted into the database when it is created for the first time.

For employees we should have the following information stored

- John Smith works in the IT department has a salary of 35,000 and extension number 1234
- John Doe works in the HR department and has a salary of 45,000 and extension number of 4164
- Traci Simms works in the Engineering department and has a salary of 50,000 and extension number of 4455
- Caitlyn Terrell works in the IT department has a salary of 40,000 and extension number 1234
- Montgomery Burns works in the CEO department and has a salary of 100,000 and extension number 0666
- Alphonso Peck works in the Engineering department and has a salary of 50,000 and extension 4455
- Lanford Burnham works in the Engineering department and has a salary of 60,000 and extension 4454

For departments we should have the following information stored

- The CEO department is in the main building and has no manager
- The HR department is in the main building and has Montgomery Burns as a manager
- The IT department the main building and has Caitlyn Terrell as a manager
- The Engineering department is in the workshop and has Lanford Burnham as a manager

For buildings we should have the following information stored

- The main building
- The workshop

Notes:

You will be required to submit a pdf and your database inside an archive and submit to the moodle by Sunday 1st at 23:55. However, this should be easily doable within the two lab sessions that you have. If you have your MySQL account at this stage you should use MySQL to generate your databese and for submission you will need to submit a SQL dump file as evidence that you have produced your database. You will also be required to include your ER diagram in your PDF. Generate the database as shown below and answer the following questions about it

Task List:

01) create an entity relationship diagram that lists the attributes of the entities mentioned above and the relationships between them. Show your diagram in your PDF (20%)

02) create a database out of your ER diagram and show the create table statements that were used. Note you are required to use and enable foreign key references in the some of the tables. (20%)

03) insert all of the data into the relevant tables (10%)

04) write a query that shows the managers of all departments and show the result (10%)

05) write a query that totals the salary's of all employees by department and order them by highest total first show the result as well (10%)

06) write a query that picks out the names of all managers and the departments that they manage show your result as well (10%)

07) write a query that picks out the names of all departments in the main building. Show your result as well (10%)

08) write a query that picks out all the extension numbers of the engineering department. Show your result as well (10%)

RELATIONAL DATABASES PRACTICAL 5: MORE EXAMPLES OF ER DIAGRAMS AND DATABASE DESIGN

Introduction:

In this practical you will be tasked with building a simple database from the spec that you will be given below. You are required to model this with an ER diagram (which must be included in your submission) and then convert this ER diagram into a database model. After this you will be expected to run some queries on your data and produce some useful information from it.

A music enthusiast website is trying to keep track of all the music that has been created. In their database they would like to store information about the artists, their albums, tracks and record labels in their database. However, as they are not database experts they have asked you to construct a database for them. You are required to design this database and test it. This is the specification that you have been given:

An artist denoted by a name can have multiple albums along with a time they started recording and a time they retired, each album recorded by an artist can have multiple tracks, however we are expected to maintain information about when the album was released and the record label(s) that published it. Each track on an album needs to have a name and also a track length (in seconds)

Notes:

You will be required to submit a pdf and your database inside an archive and submit to the moodle by Sunday 8th at 23:55. However, this should be easily doable within the two lab sessions that you have. If you have your MySQL account at this stage you should use MySQL to generate your databese and for submission you will need to submit a SQL dump file as evidence that you have produced your database. You will also be required to include your ER diagram in your PDF. Generate the database as shown below and answer the following questions about it

Task List:

01) create an entity relationship diagram that lists the attributes of the entities mentioned above and the relationships between them. Show your diagram in your PDF (20%)

02) create a database out of your ER diagram and show the create table statements that were used. Note you are required to use and enable foreign key references in the some of the tables. (10%)

03) insert information about Foo Fighters self titled debut Album and their most recent album Sonic Highways into the database (you do not need to show your commands here). **Note take advantage of your command history here as there are 20 tracks here.** (20%)

04) insert information about an album of your choice (no need to show commands here) (10%)

05) write a query that calculates the total length of all albums in seconds and show the result order them by shortest album first (10%)

06) write a query that picks out the longest track on each album by time (use the max() function here) (10%)

07) write a query that shows the names of all albums in the database along with the artist that wrote them and the record labels that published them (due to the nature of the database design do not be surprised if you multiple rows for albums with more than one label) (10%)

08) write a query that shows when each album was released (10%)

RELATIONAL DATABASES PRACTICAL 6: TAKING AN EXISTING DATABASE AND NORMALISING IT.

Introduction:

The goal of this practical is very simple. You will get a database file that contains an unnormalised database. Your goal is to figure out and turn this database into a database that satisfies 3NF. Pay close attention to the data within the database and try to figure out the relationships between them. Once you have figured out the relationships between the data and also how to normalise it, you must then reconstruct the database in 3NF. You will have free choice of primary keys and foreign keys.

Notes:

You will be required to submit a pdf and your database inside an archive and submit to the moodle by Sunday 22nd at 23:55. However, this should be easily doable within the two lab sessions that you have. If you have your MySQL account at this stage you should use MySQL to generate your databese and for submission you will need to submit a SQL dump file as evidence that you have produced your database.

Task List:

01) In your PDF show the iterative decomposition of this database into third normal form. You are not required to write out all the data in your database on each decomposition. You are required to analyse your database after each decomposition and determine what normal forms each table is in. If you decompose a table you must state your reasoning for doing so (50%)

02) regenerate the database according to the normalisation you have done above. Make sure you enforce your primary key and foreign key restrictions

RELATIONAL DATABASES PRACTICAL 7: INTRODUCTION TO THE BASICS OF MYSQL/MARIADB

Introduction:

In this practical you will be introduced to using MySQL/MariaDB. MySQL was originally an independent open source project but is now under Oracle ownership. MariaDB is a fork of MySQL so most of the functionality will remain the same for now. You can use either for the purposes of this assignment.

Here you will be shown a lot of statements for MySQL. You will be expected to run them and figure out what they do. The idea here is to get you to learn the difference between SQLite and MySQL.

Notes:

You will be required to submit a pdf and your database inside an archive and submit to the moodle by Sunday 29th at 23:55. However, this should be easily doable within the two lab sessions that you have.

How to connect to MySQL:

- In order to connect to MySQL you will need a **username** and **password** (This information has been passed to you already).
- If you are using Linux/Mac OS then run **Terminal**. If you are using Windows OS then run **Putty** (software should be installed on all PCs in the lab). If you are using your own window based machine you should install **Putty** from www.putty.org[1] .
Remember: In order to access your MySQL account you need a wired connection to gcd network. Furthermore, these accounts are only accessible within Griffith college.
- If you are using **Putty** type this command into the host name field: **username@knuth.student.gcd.ie** (replace the **username** with username you have been given). Now click **Open.**
- If using Linux/Mac OS type this into the terminal:
ssh username@knuth.student.gcd.ie (replace the **username** with username you have been given) and hit **Enter**.
- You will be asked for a **password**. Write the **password** for your username and then press **Enter**. If username and password were correct you will see a **Welcome to the Ubuntu Server!** message.
- Now type **mysql –p** and hit **Enter**. Enter the same **password** again and hit **Enter**. If you have run the commands rightly, you will see your cursor stopped at **mysql>**
- To see a list of databases, type the following: **show databases;**
- In order to use the database that is already created with the same name as your username type **use databaseName;** replace **databaseName** with your **username**. For example: **use s1010101;** and then press **Enter**.
- Now write **show tables;** and press Enter. If this is the first time and you have not created any tables, the command will return **Empty set** which means we do not have tables in the database yet.

There are also some useful commands that you may wish to have in order to query things about the mysql system to help you navigate your way around.

[1] Last accessed 24/07/2018

- SHOW DATABASES: shows you a list of databases that you have created in the MySQL system
- USE <database>: tells mysql to use a given database for running all queries against. Replace <database> with the name of the database you wish to use.
- SHOW TABLES: shows you the tables in the current database similar to .table in SQLite
- DESCRIBE <table>: shows you the schema that was used to create the given table. Similar to .schema in SQLite. Replace <table> with the name of the table that you want described.

Task List:

For each of the following write what result you get and what you think the result means.
01) SELECT VERSION(), USER();
02) SELECT CURRENT_DATE;
03) USE <database name> (if you are on the server then you only have one choice. If you are using your own MySQL/MariaDB installation put in test here)
04) CREATE TABLE pet (name VARCHAR (20), owner VARCHAR (20), species VARCHAR (20), sex CHAR (1), birth DATE, death DATE);
05) SHOW TABLES;
06) DESCRIBE pet;
07) SELECT * FROM pet;
08) INSERT INTO pet VALUES ('fluffy', 'harold', 'cat', 'F', '2008-02-04', NULL), ('claws', 'gwen', 'cat', 'M', '2009-03-17', NULL), ('buffy', 'harold', 'dog', 'F', '2004-05-13', NULL), ('fang', 'benny', 'dog' , 'M', '2005-08-27', NULL), ('bowser', 'diane', 'dog', 'M', '1994-08-31', '2010-07-29'), ('chirpy', 'gwen', 'bird', 'F', '2012-09-11', NULL), ('whistler', 'gwen', 'bird’, NULL, '2012-12-09', NULL), ('whistler', 'gwen', 'bird’, NULL, '2012-12-09', NULL);
09) SELECT * FROM pet;
10) INSERT INTO pet VALUES ('puffball', 'diane', 'hamster', 'F', '2010-03-30', NULL);
11) SELECT * FROM pet;
12) UPDATE pet SET birth = '2006-08-31' WHERE NAME = 'bowser';
13) SELECT * FROM pet;
14) SELECT * FROM pet WHERE birth >= '2011-1-1';
15) SELECT * FROM pet WHERE species = 'dog' AND sex = 'F';
16) SELECT * FROM pet WHERE species = 'snake' OR species = 'bird';
17) SELECT * FROM pet WHERE (species = 'cat' AND sex = 'M') OR (species = 'dog' AND sex = 'F');
18) SELECT name, birth FROM pet;
19) SELECT owner FROM pet;
20) SELECT DISTINCT owner FROM pet;
21) SELECT name, birth FROM pet ORDER BY birth;
22) SELECT name, birth FROM pet ORDER BY name;
23) SELECT name, birth FROM pet ORDER BY name DESC;
24) SELECT owner, COUNT(*) FROM pet GROUP BY owner;
25) SELECT species, sex, COUNT(*) FROM pet GROUP BY species, sex;

RELATIONAL DATABASES ASSIGNMENT: GENERATING A DATABASE TO MAP THE 2015 F1 SEASON.

Introduction:

The goal of this assignment is to combine all of the skills you have learned in the last 7 practical worksheets to generate a database to map information about the 2015 Formula 1 season. You will have three weeks to do this assignment because this database will be signficantly larger than those you have encountered so far. Here you will need to map, drivers, teams, tracks, results within the database. You will be required to construct it and insert information into it. All information you will require can be found here. www.formula1.com/ [2]

Due to the sheer scale of data involved you will only be required to map information about 4 drivers in particular: Lewis Hamilton, Nico Rosberg, Sebastian Vettel, and Kimi Raikkonen along with their respective teams Mercedes and Ferrari. The spec you need is listed below. NOTE: you are required to make a MySQL database for this assignment.

Specification:

In the 2015 F1 season that ended on the 29th of November there were 10 teams competing in 19 races for both a drivers championship and a constructors championship. Each team was requried to enter two cars in each and every race of the season. Each car was driven by one driver in a race. Drivers were required to set a grid in qualifying on the Saturday to for the Sunday, with the fastest driver first and the slowest last. After the race on the Sunday points would be awarded to both drivers and teams based on drivers finishing in the top 10 places. Note that sometimes drivers would retire from races or not start. In these cases you are permitted to use the value of 21 to represent these cases.

This process would be done for 19 races with the first starting in Australia and the last finishing in Abu Dhabi. Your database will be required to map the results for both qualifying and the race for each individual race in the calendar in order to answer questions about the overall season.

Notes:

You will be required to submit a pdf and your database inside an archive and submit to the moodle by Friday 18th at 17:00. However, this should be easily doable within the two lab sessions that you have. You will be required to submit a MySQL dump along with a PDF containing all of your diagrams and the normalisation process in your database along with all of your answers

Most importantly is that you must realise that there are multiple ways of generating such a database. There is no single one answer to this assignment. You have full free reign over the structure of the database. I will not give advice on what primary keys, foreign keys, table divisions, or information you will need in your database. You may ask me about the specification but it is you who must decide how to construct your database.

[2] Last accessed 24/07/2018

Depending on how you design your database you may make it easy or hard to insert information and run your queries.

Task List:

01) Generate an ER diagram of this database that maps all of the necessary concepts required for this database. You are required to map all of the relationships between all entities. (20%)

02) Normalise your database and explain why it is in third normal form. (20%)

03) insert all of the information for the four drivers and two teams listed above for 19 qualifying sessions and 19 races. You are not required to show all of your insert statements here. (10%)

04) write queries and show your results for the following (50%, 5% for each one.)

- How many times did each driver appear on the podium (top three places).
- How many points did each driver score in the season.
- How many points did each team score in the season.
- What was the average finishing position of each driver in the season.
- How many pole positions did each driver
- How many pole positions did each team get
- How many race wins did each driver get.
- How many race wins did each team get.
- The names of the races won by Sebastian Vettel and Kimi Raikkonen
- The names of the races where Lewis Hamilton and Nico Rosberg got pole position.

GRIFFITH COLLEGE DUBLIN

QUALITY AND QUALIFICATIONS IRELAND
EXAMINATION

RELATIONAL DATABASES

Lecturer(s):

External Examiner(s):

Date: **XXXXXXXX** **Time:** **XXXXXXX**

THIS PAPER CONSISTS OF SIX QUESTIONS
FOUR QUESTIONS TO BE ATTEMPTED
SECTION A – TWO QUESTIONS TO BE ATTEMPTED
SECTION B – TWO QUESTIONS TO BE ATTEMPTED
ALL QUESTIONS CARRY EQUAL MARKS
IN ALL CASES, CANDIDATES SHOULD READ THE ENTIRE QUESTION, BEFORE ANSWERING ANY PART

SECTION A – TWO QUESTIONS TO BE ATTEMPTED

QUESTION 1

Answer all parts

Central Repairs is a new motor repairs facility set up this year. It needs to store data on all cars and owners that pass through its premises as well as details of all mechanics that work there. It also has to keep records on work carried out that may have been issued to each vehicle. The repairs facility wishes to store the following data about the Owners, Owner ID, Owner name, address, and contact number. For each mechanic, it wishes to store ID number, name, and year of qualification, address and contact number. For each car it requires the following information to be stored its registration number colour, engine capacity, and manufacturer. For each Repair, it requires to store Repair ID, Fixed/Not Fixed, and Cause of fault.

When the car is repaired, an invoice of work completed is assigned to the repair. Thus only one repair can generate one invoice and each invoice is generated by one repair. If the car cannot be repaired, no invoice will be created. For each Invoice, it stores Invoice ID, Invoice Date, Invoice Details and Amount. Each invoice needs to be signed off by an employee. There are a number of mechanics working on a particular car repair and each mechanic could work on one or more than one cars each day.

(a) List the entities present in the above specification.

(2.5 marks)

(b) List the relationships that exist between these entities?

(2.5 marks)

(c) Draw an Entity Relationship (ER) diagram to represent the above scenario.

(10 marks)

(d) Develop a relational schema for your ER diagram.

(10 marks)

Total (25 marks)

QUESTION 2

The table below shows the results from a software testing laboratory. An *experiment* can be carried out on a piece of *software* multiple times and an experiment is always assigned a single *scientist*.

(a) Propose a valid primary key for the table in its current form.

software	*experiment*	*result*	*scientist*	*completionTime*
1	1	pass	Marco	06/11/2017 17:20:21
1	2	pass	Marco	06/11/2017 17:30:40
1	3	pass	Marco	06/11/2017 17:40:02
2	1	pass	Paulo	06/13/2017 12:30:00
2	2	fail	Paulo	06/15/2017 11:20:01
3	1	fail	Marco	06/21/2017 09:20:30

(5 marks)

(b) How would you reduce redundancy in the schema? Include your reasoning in your answer.

(10 marks)

(c) Show your restructured schema including data.

(10 marks)

Total (25 marks)

QUESTION 3

(a) Taking into account, union / intersection between two relations, for instance relation R and S, what conditions must be fulfilled before one can have R ∩ S or R U S.

(4 marks)

(b) Which of these (union, intersection and difference) relational algebra operators are associate and commutative?

(2 marks)

(c) What is Duplicate elimination in the context of Projection?

(4 marks)

(d) Assume the following relation EMPLOYEE has the following tuples:

Name	Office	Department	Rank
Smith	400	CS	Assistant

Jones	220	Econ	Adjunct
Green	160	Econ	Assistant
Brown	420	CS	Associate
Smith	500	Fin	Associate

Write the relational algebra expressions to

(i) Select employees in the CS department.

(3 marks)

(ii) Select only those Employees with last name Smith and who are Assistant Professors.

(3 marks)

(iii) Select only those Employees who are either Assistant Professors or work in the Economics department.

(3 marks)

(iv) List the names and departments of all employees.

(3 marks)

(v) Show the names of all employees working in the CS department.

(3 marks)

Total (25 marks)

SECTION B – TWO QUESTIONS TO BE ATTEMPTED

QUESTION 4

Answer (a), (b) and (c)

Consider the following relational schema for a training college:

student(studentID, firstName, lastName, DoB, major, level)

class(classID, meetsAt, roomNo, *teacherID*)

enrolled(*studentID*, *classID*)

teacher(teacherID, firstName, lastName, *departmentID*)

department(departmentID, name, campus)

The primary keys are underlined. The foreign keys are highlighted in blue colour. A student is uniquely identified by a student ID. Classes are uniquely identified by their ID. Students are enrolled in classes (see enrolled table). Classes are taught by teachers. One teacher can teach many classes and each teacher is affiliated with a certain department.

Write SQL queries to answer the following:

(a) Write statements to create all the tables from relational schema described above. Select appropriate data types for each attribute. Remember to set the primary and foreign keys.

(15 marks)

(b) List the IDs and full names of students on level 4 whose major is "Business" and are enrolled in more than two classes.

(5 marks)

(c) List the names of all the teachers who are working in the "Computing Department" located in "Dublin" campus.

(5 marks)

Total (25 marks)

QUESTION 5

Answer all parts

Consider the following relational schema for a company:

country(countryID, countryName, *RegionID*)

employee(employeeID, firstName, lastName, email, phoneNo, hireDate, salary, *jobID*, *managerID*, *departmentID*)

job(jobID, jobTitle, minSalary, maxSalary)

locations(locationID, address, *countryID*)

department(depatmentID, departmentName, *managerID*, *locationID*)

region(regionID, regionName)

The primary keys are underlined. The foreign keys are highlighted in italics. Each region has many countries, while each country has many (company) locations. Each location has many departments and each department has many employees working in it. Many employees can work under each job title.

Write SQL queries to answer the following:

(a) Write a query to display the full name and hire date for all employees who were hired in year 1998.

(5 marks)

(b) Write a query to get the number of employees with the same job.

(5 marks)

(c) Write a query to get the average salary for all departments employing more than 10 employees.

(5 marks)

(d) Write a query to find the full name and salary of the employees who earn more than 50000 and work in any of the IT departments.

(5 marks)

(e) Write a query to get the department name and number of employees in the department.

(5 marks)
Total (25 marks)

QUESTION 6

Answer all parts

(a) What is an Object Oriented Database? How is it different from Relational Databases?

(4 marks)

(b) What is the definition of distribution transparency in the context of Distributed Database Systems? Define Concurrency and Failure Transparencies.

(9 marks)

(c) List three advantages and three disadvantages of the Distributed Database Systems.

(12 marks)
Total (25 marks)