



FIT9132 Introduction to Databases

2020 Semester 1

Assignment 2 - SQL - Traffic Demerit System (TDS)

Assignment weighting 15% - Lecturer in Charge: Manoj Kathpalia

The local state government wishes to develop a database to record the details of "on-the-spot" traffic offences incurred by drivers. When a driver commits an offence they are stopped and booked "on the spot" by a police officer and provided with an offence notice. Offences cause drivers to incur demerit points which accumulate against the drivers' licence.

The system records the details of all registered vehicles driven in the local government area. A vehicle is identified by a Vehicle Identification Number (VIN). Background reading on a VIN is available from several sites such as [AutoCheck](#). Other attributes recorded for a vehicle are the vehicle's year of manufacture, main colour, model name, manufacturer name and vehicle type.

Drivers are identified by their licence number. Other attributes recorded for a driver are the driver's mobile number, first name, last name, residential address, date of birth and expiry date for the licence. It is *assumed* for this task that each driver has a mobile number and that number will be unique.

Each demerit (potential driving offence) is identified by a demerit code. A description of the demerit and the number of points incurred for that demerit are also recorded. For example, the demerit with code 108, has a description of "Failing to give way, or stop, or remain stopped" and results in 3 demerit points being incurred by the driver.

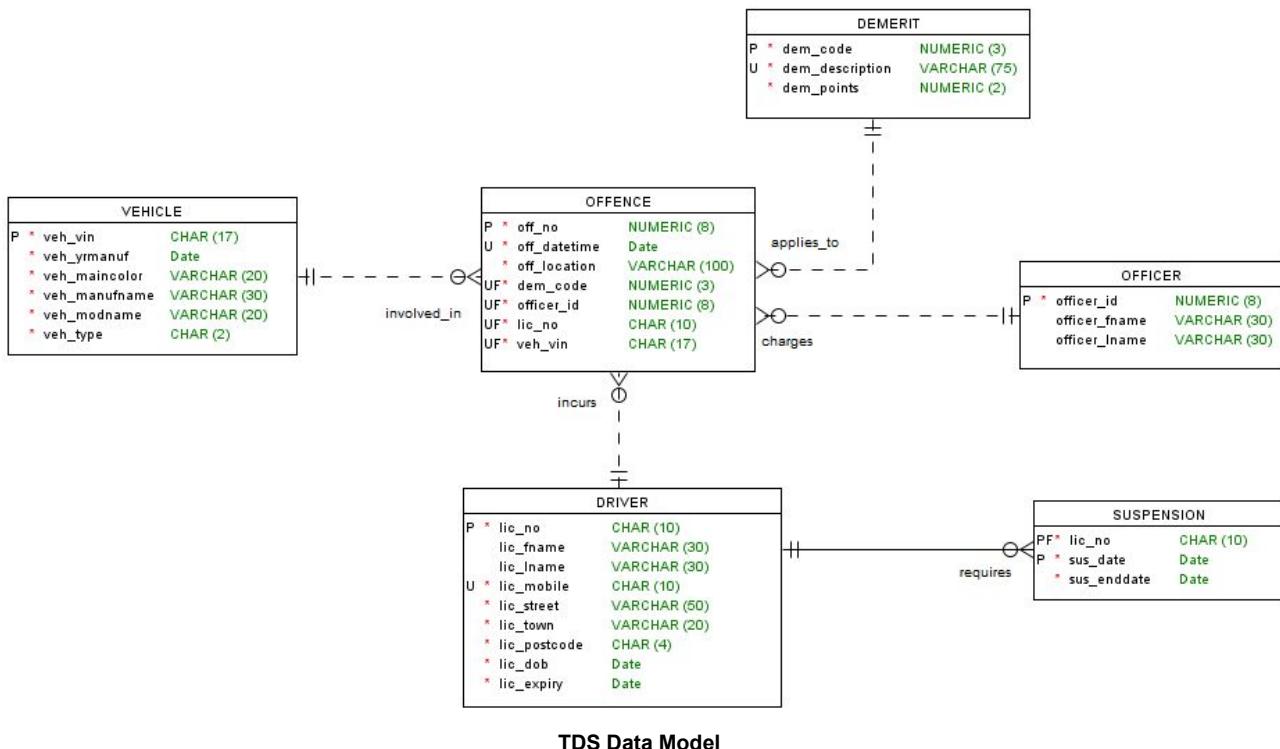
Each traffic offence by a driver is identified by an offence number. The location of the offence and the date and time at which the offence occurred is also stored. In some circumstances when a driver is stopped by a police officer, for example, a speeding offence, the police officer may also identify other offences such as "Driving contrary to a major defect notice". Each offence recorded has a unique offence number and is for only one particular demerit. The police officer who issued the offence is also recorded.

Once a driver has accumulated 12 demerit points over a period of three years the driver's licence is suspended for a period of 6 months. When an offence committed by a driver is added to the system by the police officer, the system at that time will determine automatically if the licence should be suspended or not. When this occurs the driver's licence is suspended from the date recorded for the offence which caused the points to equal or exceed the permitted maximum points. The date at which the suspension began and the date at which it ends are to be stored in the system. The driver, if suspended, is not allowed to drive and driving while suspended will lead to a criminal charge and may cause the driver to be jailed.

After a driver's suspension ends, their total demerit points are reset to zero and they begin accumulating points again. For any previously suspended driver, to determine the new suspension date, the system will calculate the accumulated demerit points from the date their last suspension ended.

Police officers are identified by an officer id. The system only records the officer's first name and last name.

A data model has been created for Traffic Demerit System and is shown below (it is also available from Moodle):



For this assignment, you will populate these tables with appropriate test data and write the SQL queries specified below. You must ensure that any activities you need to carry out in the database to complete the assignment conform to the requirements of the data model displayed above.

The schema/insert file for creating this model is available in the archive ass2-student.zip - this file creates the Traffic Demerit System tables and populates several of the tables - you should read this schema carefully and be sure you understand the various data requirements. **You must not alter the schema file in any manner, it must be used as supplied.** This schema file contains a single commit after the inserts have completed since this is setting up an initial state of the database for you to work with, you should not use this as your method of approach.

IMPORTANT points for you to observe, when completing this assignment, are:

1. The ass2-student.zip archive also contains four SQL scripts for you to code your answers in, **you should ensure these files are regularly pushed to GitLab server so a clear development history is available for the marker to verify (a minimum of four pushes are required).** In each file, you **must** fill in the header details with your name and student ID before beginning any work. **Your script files must not include any SPOOL or ECHO commands.** Although you might include such commands when testing your work **they must be removed before submission** (a 5 mark grade penalty will be applied if your documents contain spool or echo commands)
2. You are free to make assumptions if needed. However, **your assumptions must align with the details here and in the assignment 2 forum** and must be clearly documented (see the required submission files).

REMEMBER you must keep up to date with the Moodle assignment 2 forum where further clarifications may be posted (this forum is to be treated as your client).

Please be careful to ensure you do not post anything which includes your reasoning, logic or any part of your work to this assignment forum as doing so violates Monash plagiarism/collusion rules.

3. Queries that use subqueries and SQL conditions **unnecessarily** to get required data will be **penalised**. Views **must not** be used in arriving at any solutions for the tasks you are required to complete as part of this assessment.
4. In handling dates, the default date format must not be assumed; **you must make use of the TO_DATE and TO_CHAR functions in all date situations.** Failure to do so will incur a 50% grade penalty for questions involving dates.
5. In completing the following tasks, you should **design your test data so that you always get output for the SQL scripts/queries specified below** - this may require you to add further data as you move through completing the required tasks. **Queries that are correct but do not produce any output ("no rows selected" message) using your test data will lose 50% of the marks allocated,** so you should carefully check your test data and ensure it thoroughly validates your SQL queries.

You may need to **rerun the schema**, especially when you have been experimenting with your solutions and may have corrupted the database unintentionally. If you suspect that there might be such problems, simply rerun the schema. The schema includes the appropriate drop commands at the head of the file.

Assignment Tasks

Using the supplied schema file (`tds_ass2_schm_insert.sql`) create the tables for the Traffic Demerit System and insert the supplied values. This provides you with a *starting point* for the following tasks.

TASK 1: Data Manipulation (23 marks):

(a) Load **selected tables** with your own additional test data using the supplied `Q1a-tds-insert.sql` script file, and SQL commands which will insert, as a minimum, the following sample data -

- 20 OFFENCES
- 3 SUSPENSIONS

Please note, these are the *minimum number of entries you must insert*; you are encouraged to insert more to provide a richer data set to draw from.

For this task **only**, data that you add in the database should follow the rules mentioned below:

1. The primary key values for this data should be hardcoded values (i.e., **NOT** make use of sequences) and must consist of values below 100.
2. Offences that you add must involve at least 12 different drivers.
3. You **must not** add any offences for the following driver:
 - Lion Lawless of 72 Aberg Avenue Richmond South 3121 (Licence no.: 100389)
4. Dates used must be chosen between the 1st January 2016 and 30th June 2019.
5. For each year from 2016 to 2019, you should add at least 4 offences.
6. The suspensions you add in the database should not be in just one year, e.g., you should not add all the suspensions in 2018.
7. The suspensions you add in the database should not belong to just one driver, e.g., you should not add all the suspensions for the driver with licence number 100100.
8. A suspended driver cannot incur extra offences under this system while suspended. If a driver commits a driving offence while suspended their vehicle is impounded and they are sent before the court (this means you must not add an offence to a suspended driver until their suspension period has expired).

9. The supplied trigger shows the current total demerit points for a driver **when you insert a new offence**. You will need to run your inserts one by one in your offence date order, taking appropriate action if the drivers' accumulated points pass the 12 demerit point threshold. Any offence that you add, and any appropriate action that you may have to take as a result of the addition of the new offence, must be managed as a single transaction.

For this task **ONLY**, you can look up and include values for the loaded tables/data directly where required. However, if you wish, you can still use SQL to get any non-key values.

You are reminded again that in carrying out this task you must not modify any data or add any further data to the tables which were previously populated by the supplied schema file.

[14 marks]

For all subsequent questions (Q1b onwards) **you are NOT permitted to manually:**

- lookup a value in the database, obtain its primary key or the highest/lowest value in a column, or
- calculate values external to the database, e.g., on a calculator and then use such values in your answers.

You must ONLY use the data as provided in the text of the questions. Where a particular case (upper case, lower case, etc.) for a word is provided you **must only use that case**. You may divide names such as Zora Mandrey into the first name of Zora and a last name of Mandrey if required. **Failure to adhere to this requirement will result in a mark of 0 for the relevant question.**

At any point in your solution, *when inserting an offence* you must ensure that

```
SET SERVEROUTPUT ON;
```

has been run in your SQL session so that you receive visual feedback from the checkPointsAccumulated trigger.

(b) For the following tasks, **your SQL must correctly manage transactions and use sequences to generate new primary keys for numeric primary key values** (under no circumstances may a new primary key value be hardcoded as a number or value). Your answers for these tasks must be placed in the supplied SQL Script

Q1b-tds-dm.sql

You are reminded again that queries that use subqueries and SQL conditions unnecessarily to get required data will be penalised.

- (i) Create a sequence which will allow entry of data into the OFFENCE table - the sequence must begin at 100 and go up in steps of 1 (i.e., the first value is 100, the next 101, etc.)

[1 mark]

- (ii) Lion Lawless of 72 Aberg Avenue Richmond South 3121 (Licence no.: 100389) has been very inconsiderate of others on the road over the years and has committed several offences that have been booked by highly vigilant TDS officers at various different locations. Lion Lawless was riding the same motorbike, a 1994 Red Yamaha FZR600 (JYA3HHE05RA070562) at the time of committing these offences. Lion Lawless has only committed the offences listed below. The details of the bookings for Lion are as follows:

- 10-Aug-2019 08:04 AM booked for traffic offence "Blood alcohol charge" by police officer Dolley Hedling (10000011)
- On 16-Oct-2019 9:00 PM booked for traffic offence "Level crossing offence" by police officer Geoff Kilmartin (10000015)
- On 7-Jan-2020 7:07 AM booked for traffic offences "Exceeding the speed limit by 25 km/h or more" by police officer Geoff Kilmartin (10000015)

Take the necessary steps in the database to record these offences.

[5 marks]

- (iii) Lion Lawless of 72 Aberg Avenue Richmond South 3121 (Licence no.: 100389) had appealed against the "Exceeding the speed limit by 25 km/h or more" offence he has been alleged to have committed on 07-Jan-2020 at 7:07 AM.

After careful consideration and taking into account that speed guns at times are not very accurate, TDS has decided to lessen the offence to "Exceeding the speed limit by 10 km/h or more but less than 25 km/h" but has strongly warned Lion Lawless to be more careful in future. Take the necessary steps in the database to record this change.

[3 marks]

TASK 2: SQL Queries (55 marks):

Your answers for these tasks must be placed in the supplied SQL Script
Q2-tds-queries.sql

ANSI joins must be used where two or more tables are to be joined, under no circumstances can "implicit join notation" be used - see the week 7 workshop slide 22 and tutorial

- (i) Show the demerit points and demerit description for all the demerits that either contains the word “heavy” or “Heavy” or start with the word “Exceed” in the description. The column headings in your output should be renamed as Demerit Points and Demerit Description. The output must be sorted in ascending format by demerit points and where two demerits have the same points sort them in ascending format of demerit description. Your output must have the form shown below.

Demerit Points	Demerit Description
1	Exceed speed limit by less than 10km/h
3	Driving a fatigue-regulated heavy vehicle against max work req. - Severe
3	Driving a fatigue-regulated heavy vehicle against min. rest req. - Severe
3	Driving a fatigue-regulated heavy vehicle while impaired by fatigue
3	Exceeding the speed limit by 10 km/h or more but less than 25 km/h
4	Driving a fatigue-regulated Heavy vehicle against max work req. - Critical
4	Driving a fatigue-regulated Heavy vehicle against min. rest req. - Critical
4	Exceeding the speed limit by 25 km/h or more

[4 marks]

- (ii) For all “Range Rover” and “Range Rover Sport” models, show the main colour, VIN and manufacture year for all the vehicles that were manufactured from 2012 to 2014. The column headings in your output should be renamed as Main Colour, VIN and Year Manufactured. The output must be sorted by manufacture year in descending format and where more than one vehicle was manufactured in the same year sort them by colour in ascending format. Your output must have the form shown below.

Main Colour	VIN	Year Manufactured
Black	SALWG2VFXEA308415	2014
White	SALWR2TF0EA352419	2014
Charcoal	SALMF1D42CA345640	2012

[4 marks]

(iii) Show the driver licence number, full name (firstname and lastname together), date of birth, full address (street, town and postcode together), suspension start date and suspension end date for all the drivers who have had their licence suspended in the last 30 months. You need SQL to calculate 30 months from the day this query would be executed by the user. The column headings in your output should be renamed as Licence No., Driver Fullname, DOB, Driver Address, Suspended On and Suspended Till. The output must be sorted by licence number in ascending format and where there is one licence number suspended more than once sort them by suspended date in descending format. Your output must have the form shown below. Your output can clearly be different from the following output.

Licence No.	Driver Fullname	DOB	Driver Address	Suspended On	Suspended Till
100018	Garvey Nocair	20-Oct-1989	9 Anniversary Place Richmond	3121 17/JAN/18	17/JUL/18

[5 marks]

(iv) TDS would like to find out if there is any correlation between different months of a year and demerit codes so you have been assigned to generate a report that shows for **ALL** the demerits, the code, description, total number of offences committed for the demerit code so far in **any month (of any year)** and then the total of offences committed for the demerit code in **each month (of any year)**. The column headings in your output should be renamed as Demerit Code, Demerit Description, Total Offences (All Months), and then the first three letters of each month (with the first letter in uppercase). The output must be sorted by Total Offences (All Months) column in descending format and where there is more than one demerit code with the same total, sort them by demerit code in ascending format. Your output must have the form shown below. Your output can clearly be different from the following output.

Demerit Code	Demerit Description	Total Offences (All Months)	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
100	Blood alcohol charge	12	3	1	0	3	0	0	2	1	1	1	0	0
101	Level crossing offence	5	2	2	1	0	0	0	0	0	0	0	0	0
131	Following too closely	3	1	0	2	0	0	0	0	0	0	0	0	0
113	Risk colliding with exiting, boarding or waiting tram passengers	2	0	0	0	0	0	1	0	0	1	0	0	0
127	Exceed speed limit by less than 10km/h	2	1	0	0	0	0	0	1	0	0	0	0	0
128	Driving contrary to a minor defect notice	2	0	0	0	0	1	1	0	0	0	0	0	0
103	Using a visual display unit or television illegally while driving	1	0	0	0	0	0	0	0	0	0	1	0	0
124	Turn or stop without signalling	1	0	0	1	0	0	0	0	0	0	0	0	0
125	Turn improperly	1	0	0	0	1	0	0	0	0	0	0	0	0
99	Exceeding the speed limit by 25 km/h or more	0	0	0	0	0	0	0	0	0	0	0	0	0
102	Using a mobile phone illegally while driving	0	0	0	0	0	0	0	0	0	0	0	0	0
104	Driving a fatigue-regulated Heavy vehicle against max work req. - Critical	0	0	0	0	0	0	0	0	0	0	0	0	0
105	Driving a fatigue-regulated Heavy vehicle against min. rest req. - Critical	0	0	0	0	0	0	0	0	0	0	0	0	0
106	Exceeding the speed limit by 10 km/h or more but less than 25 km/h	0	0	0	0	0	0	0	0	0	0	0	0	0

[7 marks]

(v) Find out which manufacturer's vehicles are involved in the highest number of offences which incur 2 or more demerit points. Show the manufacturer name and the total number of offences that the manufacturer's vehicles are involved in. The column headings in your output should be renamed as Manufacturer Name and Total No. of Offences. The output must be sorted by Total No. of Offences column in descending format and where there is more than one manufacturer with the same total, sort them by manufacturer name in ascending format. Your output can clearly be different from the following output.

Manufacturer Name	Total No. of Offences
Nissan	7

[7 marks]

(vi) Find out the drivers who have been booked more than once for the same offence by an officer with the last name as that of their last name. Show the driver licence number, driver full name (firstname and lastname together), officer number, officer full name (firstname and lastname together). The column headings in your output should be renamed as Licence No., Driver Name, Officer ID, Officer Name. The output must be sorted by driver licence number column in ascending format.

Licence No.	Driver Name	Officer ID	Officer Name
100011	Debbie Hedling	10000011	Dolley Hedling

[7 marks]

(vii) For each demerit code for which an offence has been recorded, find out the driver/s who has/have been booked the most number of times. Show the demerit code, demerit description, driver licence number, driver full name (firstname and lastname together) and the total number of times the driver has been booked in the output. The column headings in your output should be renamed as Demerit Code, Demerit Description, Licence No., Driver Fullname and Total Times Booked. The output must be sorted by demerit code in ascending format and where for one demerit there is more than one driver booked the most number of times sort them by licence number in ascending format. Your output must have the form shown below. Your output can clearly be different from the following output.

Demerit Code	Demerit Description	Licence No.	Driver Fullname	Total Times Booked
100	Blood alcohol charge	100011	Debbie Hedling	4
100	Blood alcohol charge	100018	Garvey Nocair	4
101	Level crossing offence	100018	Garvey Nocair	2
101	Level crossing offence	100100	Daniel Palphramand	2
103	Using a visual display unit or television illegally while driving	100081	Mireielle Tufts	1
106	Exceeding the speed limit by 10 km/h or more but less than 25 km/h	100389	Lion Lawless	1
113	Risk colliding with exiting, boarding or waiting tram passengers	100001	Dalt Alfonsini	1
113	Risk colliding with exiting, boarding or waiting tram passengers	100074	Boyce Ownsworth	1
124	Turn or stop without signalling	100034	Dolphie Badriver	1
125	Turn improperly	100061	Velvet Krysiak	1
127	Exceed speed limit by less than 10km/h	100060	Annnora Golston	1
127	Exceed speed limit by less than 10km/h	100082	Abeu Hedling	1
128	Driving contrary to a minor defect notice	100044	Luisa Dunnet	1
128	Driving contrary to a minor defect notice	100064	Wilmette Goldsberry	1
131	Following too closely	100034	Dolphie Badriver	2

[8 marks]

(viii) For each region, show the number of vehicles manufactured in the region and the percentage of vehicles manufactured in the region. The last row of the output shows the totals - the second column which shows the total number of vehicles manufactured in all regions (which is the total of all the individual totals in this column) and the third column of which shows the total percentage of vehicles manufactured in all the regions (which is the total of all the individual percentages in this column). The first character of the VIN represents the region where the vehicle was manufactured. In your SQL, you need to determine the region in which the vehicle was manufactured using the following table (Source: [wikibooks](#)).

First Character of VIN	Region
From A to C	Africa
From J to R	Asia
From S to Z	Europe
From 1 to 5	North America
From 6 to 7	Oceania
From 8 to 9	South America
Anything else	Unknown

The column headings in your output should be renamed as Region, Total Vehicles Manufactured and Percentage of Vehicles Manufactured. The output must be sorted by Total Vehicles Manufactured in ascending format and where there is more than one region with the same total, sort them by region in ascending format. Your output must have the form shown below.

Region	Total Vehicles Manufactured	Percentage of Vehicles Manufactured
Oceania	1	0.47%
South America	1	0.47%
North America	54	25.59%
Asia	66	31.28%
Europe	89	42.18%
TOTAL	211	99.99%

[13 marks]

TASK 3: Design Modifications (22 marks):

Your answers for these tasks must be placed in the supplied SQL Script **Q3-tds-mods.sql**

These tasks should be attempted only after task 1 and task 2 have been successfully completed. They are to be completed on the "live" database ie. the database with the data loaded from your previous work. For this question, **you must not make use of any PL/SQL.**

(i) TDS would like to be able to easily determine the total number of times each police officer has booked a driver for a traffic offence. Add a new attribute which will record the number of times each officer has booked drivers.

This attribute must be initialised to the correct current number of times each officer has booked drivers based on the data which is currently stored in the system.

[4 marks]

(ii) The problem TDS might face with the current database is if an offence is revoked for some valid reason, it is impossible for them to keep the offence information for the revoked offence in the database. TDS would like to fix this problem such that they are able to keep information about a revoked offence along with the other **required** information such as when it was revoked, who revoked the offence and the reason for revocation. For quick access, it has been decided that a column is also required to indicate Yes or No if an offence has been revoked or not. There will always be only one reason that can be associated with a revocation. TDS will not be able to add all the reasons for a revocation of the offences initially and so your solution should allow them to add new reasons as and when there is a need. At this stage, the only reasons for revocation of an offence TDS is interested in recording are First offence exceeding the speed limit by less than 10km/h (FOS), Faulty equipment used (FEU), Driver objection upheld (DOU), Court hearing (COH), and Error in proceedings (EIP). Each reason code will have exactly 3 letters.

When providing the solution, you must consider the following:

- a. only around 1% of offences recorded are revoked and there can be hundreds of thousands of offences in the offence table,
- b. all existing offences, after implementing this requirement, must automatically be recorded as not revoked,
- c. appeals are made by completing a paper form and sending the completed form to TDS either by fax or mail and, at this stage, there is *no need* to record unsuccessful appeals in the database.

[6 marks]

(iii) TDS has found that having just the vehicle's main colour in the database, in some cases, is inadequate in helping identify a vehicle that has been involved in a traffic offence. As a consequence, they have decided to now also record if any outer part other than the body of the vehicle is of a different colour. At this stage, the only other parts TDS is interested in recording the colour of are the Spoiler (SP), Bumper (BM) and Grilles (GR) but this may change if the need arises and so it should be able to be changed easily. Each outer part code will have exactly 2 letters. **Where the colour of grilles, spoiler (if any) or Bumper is the same as that of the body colour, there is no need to record anything in the database.**

TDS is also intending to find a solution such that new colours can be easily introduced when a manufacturer releases a new colour. Each colour must be identified by a unique colour number (which is auto-generated by the system) and has its colour description recorded as shown below (sample data only shown).

80	Blue
81	Red

This data must be collected from the current state of the vehicle table via SQL only (you cannot assume the presence of any particular colour). Subsequent colours will be added directly to this collection via INSERT statements. **You are reminded again that you must not make use of any PL/SQL in solving this task.**

At this stage, the vehicles that have different colours for some of the above-mentioned outer parts are as follows:

Vehicle 1:

VIN: ZHWEF4ZF2LLA13803

Manufacturer Name: Lamborghini

Model Name: Huracan EVO

Vehicle main colour: Grey

Spoiler colour: Black

Bumper colour: Grey

Grilles colour: Magenta

Vehicle 2:

VIN: ZHWES4ZF8KLA12259

Manufacturer Name: Lamborghini

Model Name: Huracan Performante

Vehicle main colour: Black

Spoiler colour: Yellow

Bumper colour: Blue

Grilles colour: Black

Change the database to meet these requirements. Note: You should only use the necessary DDL and DML statements to achieve these requirements.

[12 marks]

SUBMISSION REQUIREMENTS

Due Date: Friday 12th June 2020 at 5 PM (Week 12)

*Please note, if you need to resubmit, you **cannot** depend on your tutors' availability, for this reason, please be **VERY CAREFUL** with your submission. It is strongly recommended that you submit several hours before this time to avoid such issues.*

For this assignment there are four files you are **required** to submit:

- Q1a-tds-insert.sql
- Q1b-tds-dm.sql
- Q2-tds-queries.sql
- Q3-tds-mods.sql

If you need to make any comments to your marker/tutor please place them at the head of each of your solution scripts in the "Comments for your marker:" section.

Do not zip these files into one zip archive, submit four independent SQL scripts. The individual files must also have been pushed to the FIT GitLab server with an appropriate history as you developed your solutions.

Late submission will incur penalties as outlined in the unit guide.

Please note we **cannot mark any work on the GitLab Server**, you need to ensure that you submit correctly via Moodle since it is only in this process that you complete the required student declaration without which work **cannot be assessed**.

It is your responsibility to ENSURE that the files you submit are the correct files - we strongly recommend after uploading a submission, and prior to actually submitting, that you download the submission and double-check its contents.

Your assignment **MUST** show a status of "Submitted for grading" before it will be marked.

Submission status

Attempt number	This is attempt 1.
Submission status	Submitted for grading 
Grading status	Not graded

If your submission shows a status of "Draft (not submitted)" it will not be assessed and **will incur late penalties after the due date/time**.

Please **carefully** read the documentation under the "Assignment Submission" on the Moodle Assessments page which covers things such as extensions and resubmission.

Criteria for marking:

Submissions will be graded on:

- the correct application of relational database principles,
- the correct handling of transactions and the setting of appropriate transaction boundaries i.e. correct placement of commits, and
- the correct application of SQL statements and constructs to:
 - populate tables,
 - modify existing data in tables,
 - prepare reports by retrieving the required data in the required format, and
 - modify the "live" database structure to meet the expressed requirements (including appropriate use of constraints). In making these modifications there must be no loss of existing data or data integrity within the database.

Submissions will be grade penalised if they:

- contain SET ECHO ... or SPOOL commands
- make use of views
- use subqueries and SQL conditions unnecessarily,
- do not use to_char/to_date where appropriate in handling dates,
- do not have an appropriate development history on the FIT GitLab server for all source files (at least four pushes required).