MONASH University
Information Technology

FIT3171 Databases

2021 SSB

Assignment 2 - SQL

Run Monash (RM)

Learning Objectives: 2, 5, 6 (see Unit Preview)

Assignment weighting 20%

Assignment marked out of 100 and released as a grade out of 20

Run Monash (RM) is a running carnival which is held on separate dates at both Monash Caulfield and Clayton campuses during different seasons (Summer, Autumn, Winter and Spring) of a year. The carnival naming convention that Run Monash uses is *RM <season name> Series <campus name> <year>*. So, for example, a carnival to be held in Summer season at Caulfield campus in 2020 will be named as *RM Summer Series Caulfield 2020.*

Anyone can attend an RM Carnival, the carnivals are open to the public as well as Monash staff and students. A carnival is run on a particular date, in a particular location and only lasts for one day. RM *only runs one carnival on any particular date*. During a carnival a range of events are offered from the following list (only some may be offered):

- Marathon 42.2 Km
- Half Marathon 21.1 Km
- 10 Km Run
- 5 Km Run
- 3 Km Community Run/Walk

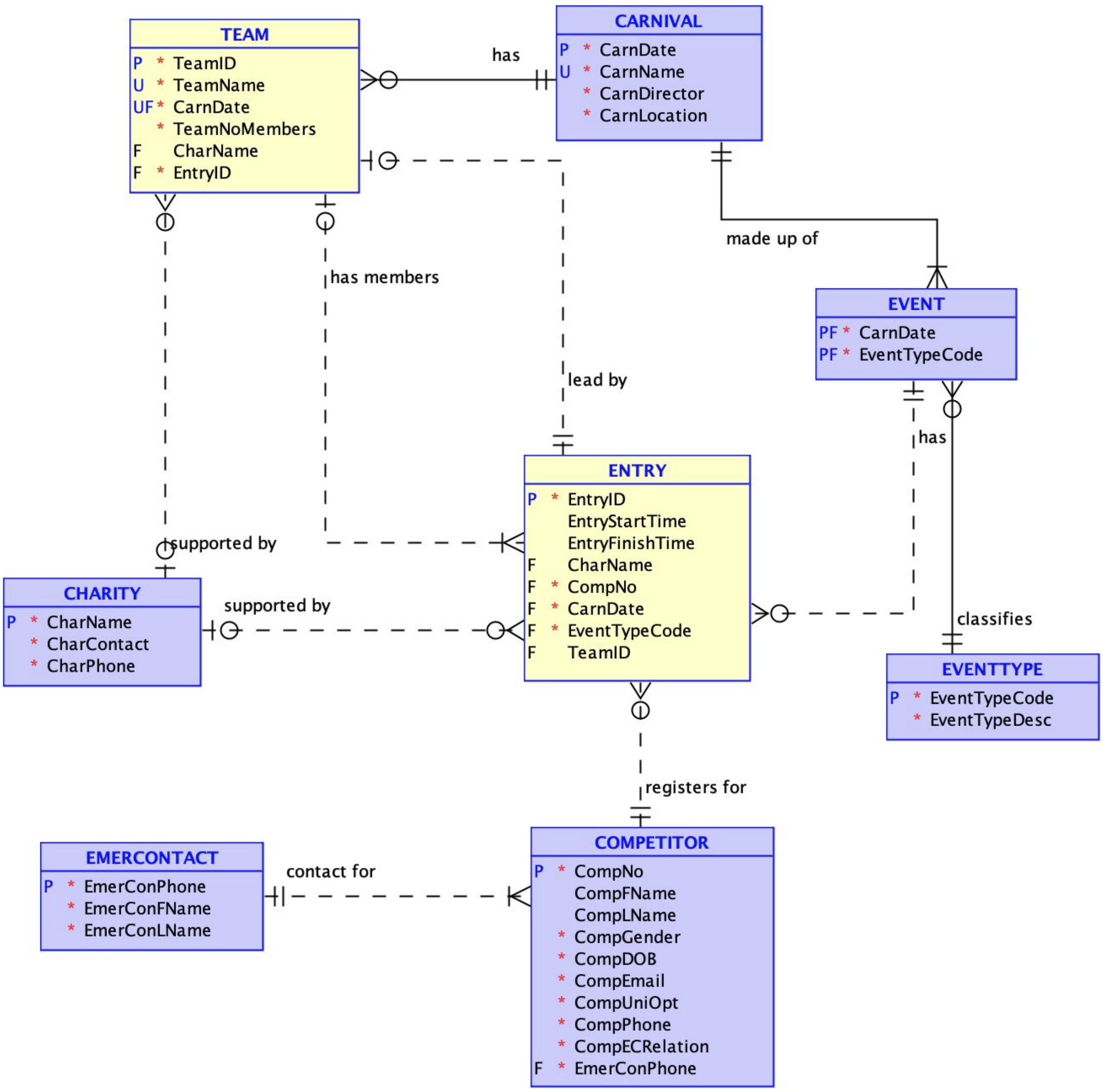
Run Monash expects to offer around 10 - 20 such events across all carnivals in a given year.

When a competitor initially registers for Run Monash, they are assigned a unique competitor number. A competitor is required to provide details of an emergency contact at the time of registration. The relationship to the competitor can be Parent (P), Guardian (G), Partner (T) or Friend (F).

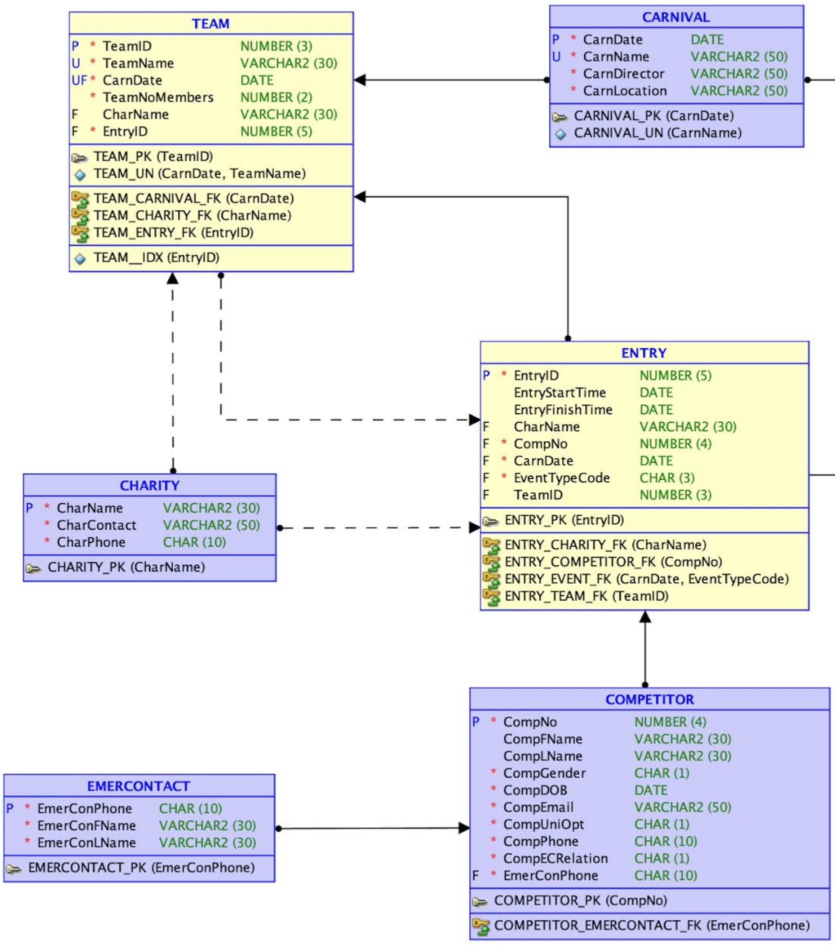
When a carnival is being offered, Run Monash contacts all registered competitors and provides details of the carnival date and what events are on offer. Competitors can enter for only one event within a carnival. Every entry is assigned a unique entry id (e.g. 3021). Using official timing devices at the carnival, Run Monash records the entrants starting and finishing times.

A major focus of the Run Monash Carnivals is to raise funds for various charities. When a competitor enters an event, they may nominate a charity for which they will raise funds (not all competitors will select a charity for each event they enter). Competitors who have entered the carnival can also form teams with other competitors, whom they know and who have entered the carnival, to support their training and run as a group. The first competitor to register a team for a given carnival is assigned as the Team Manager. Teams are identified by a unique team name which the team manager must select when they first create the team. This team manager can then add/invite other competitors from the carnival to join their team. Team names are unique *only* within a given carnival, A given team name may be reused by different competitors in a different carnival as teams are recreated for each carnival depending on which competitors have entered an event for the carnival. Run Monash wishes to record, as part of the stored data, how many members are on each team. Teams may also nominate a charity for which they will raise funds, although not all teams will do so. All charities for which funds can be raised must first be approved by Run Monash. Note that an individual competitor may be supporting a charity as an individual and also the same or a different charity as a team member.

A data model has been developed for this situation. The logical model is shown below:



From this logical model the following relational model has been created:


TEAM
P * TeamID NUMBER (3)
U * TeamName VARCHAR2 (30)
UF * CarnDate DATE
* TeamNoMembers NUMBER (2)
F CharName VARCHAR2 (30)
F * EntryID NUMBER (5)
TEAM_PK (TeamID)
TEAM_UN (CarnDate, TeamName)
TEAM_CARNIVAL_FK (CarnDate)
TEAM_CHARITY_FK (CharName)
TEAM_ENTRY_FK (EntryID)
TEAM__IDX (EntryID)
CARNIVAL
P * CarnDate DATE
U * CarnName VARCHAR2 (50)
* CarnDirector VARCHAR2 (50)
* CarnLocation VARCHAR2 (50)
CARNIVAL_PK (CarnDate)
CARNIVAL_UN (CarnName)
ENTRY
P * EntryID NUMBER (5)
EntryStartTime DATE
EntryFinishTime DATE
F CharName VARCHAR2 (30)
F * CompNo NUMBER (4)
F * CarnDate DATE
F * EventTypeCode CHAR (3)
F TeamID NUMBER (3)
ENTRY_PK (EntryID)
ENTRY_CHARITY_FK (CharName)
ENTRY_COMPETITOR_FK (CompNo)
ENTRY_EVENT_FK (CarnDate, EventTypeCode)
ENTRY_TEAM_FK (TeamID)
EVENT
PF * CarnDate DATE
PF * EventTypeCode CHAR (3)
EVENT_PK (CarnDate, EventTypeCode)
EVENT_CARNIVAL_FK (CarnDate)
EVENT_EVENTTYPE_FK (EventTypeCode)
CHARITY
P * CharName VARCHAR2 (30)
* CharContact VARCHAR2 (50)
* CharPhone CHAR (10)
CHARITY_PK (CharName)
COMPETITOR
P * CompNo NUMBER (4)
CompFName VARCHAR2 (30)
CompLName VARCHAR2 (30)
* CompGender CHAR (1)
* CompDOB DATE
* CompEmail VARCHAR2 (50)
* CompUniOpt CHAR (1)
* CompPhone CHAR (10)
* CompECRelation CHAR (1)
F * EmerConPhone CHAR (10)
COMPETITOR_PK (CompNo)
COMPETITOR_EMERCONTACT_FK (EmerConPhone)
EVENTTYPE
P * EventTypeCode CHAR (3)
* EventTypeDesc VARCHAR2 (50)
EVENTTYPE_PK (EventTypeCode)
EMERCONTACT
P * EmerConPhone CHAR (10)
* EmerConFName VARCHAR2 (30)
* EmerConLName VARCHAR2 (30)
EMERCONTACT_PK (EmerConPhone)

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 tasks conform to the requirements of the provided data model**.

The schema/insert file for creating this model (RM-schema-insert.sql) is available in the archive ass2-student.zip - this file creates the Run Monash tables and populates several of the tables (those shown in purple on the supplied model) - 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 standard method of approach.

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

1. The ass2-student.zip archive also contains five 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 marks 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 Ed Assignment 2 forum*** and must be clearly documented (see the required submission files).
REMEMBER you must keep up to date with the Ed 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 where appropriate***. Failure to do so will incur a 50% grade *penalty* for questions involving dates.

5. PL/SQL (ie, user created functions, procedures, triggers, packages or PL/SQL structures) **must not be used** for any part of the solutions to TASK 1 - TASK 3. Triggers can only be used for TASK 4.

6. In completing the following tasks, you must ***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. Such extra data MUST be added as part of Task 1A (ie. as part of your load of test data). ***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

TASK 1: Data Manipulation (20 marks):

(a) Load selected tables with your own additional test data: using the supplied **T1a-rm-insert.sql** script file, code the SQL commands which will insert, as a minimum, the following sample data into the yellow coloured relations in the logical model -

- 20 ENTRIES,
- 5 TEAMS

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. *You may treat all of the data that you add as a single transaction since you are setting up the initial test state for the database.*
2. 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.
3. The entries and teams must be spread across at least three carnivals *which have been completed*.

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.

[10 marks]

For all subsequent tasks (Task 1b 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. Any necessary calculations must be carried out as part of your SQL code.

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 Brigid Radcliffe into the first name of Brigid and a last name of Radcliffe if required. **Failure to adhere to this requirement will result in a mark of 0 for the relevant question**.

(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 **T1b-rm-dm.sql.**

(i) Create sequences (one for use per table) to provide primary keys for data entry into the following tables:

- COMPETITOR
- ENTRY
- TEAM

The sequences 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) Brigid Radcliffe, who has previously registered at RunMonash, has contacted Run Monash and indicated she would like to run as a competitor in the "RM Summer Series Clayton 2021" carnival. She would like to enter the "21.1 Km Half Marathon" event. You may assume that Brigid's phone number 1234567890, is unique in the current competitor data. She has indicated, for this carnival, that she will raise funds to support the "Amnesty International" charity.

Make these changes to the data in the database. This entire registration should be treated as a single transaction.

[3 marks]

(iii) A few weeks later Brigid Radcliffe emails the Run Monash organisers and reports that she has suffered an injury in training and would like to change her entry (called downgrading) for the "RM Summer Series Clayton 2021" carnival from the "21.1 Km Half Marathon" to the "10 Km Run".You may assume that Brigid's phone number 1234567890, is unique in the current competitor data. She also indicates that she would like to form a team called "Kenya Speedstars" for this carnival. Run Monash staff check and confirm that the team name is not currently in use for this carnival, so inform her that such a team can be created.The new team will support the "Beyond Blue" charity.

Make these changes to the data in the database. These changes must be treated as a single transaction.

[3 marks]

(iv) The following week Brigid Radcliffe emails the Run Monash organisers and reports that her injury has got a lot worse and that she will need to withdraw from the "RM Summer Series Clayton 2021" carnival. Although she has asked a few friends to join her team for this carnival she is not sure if any have actually taken up the offer; it is possible some have. She did direct that her team, "Kenya Speedstars", should be disbanded. Brigid Radcliffe indicated that she is looking forward to competing in the next 2021 carnival. You may assume that Brigid's phone number 1234567890, is unique in the current competitor data.

Make these changes to the data in the database. These changes must be treated as a single transaction.

[3 marks]

TASK 2: SQL Queries (55 marks):

Your answers for these tasks must be placed in the supplied SQL Script **T2-rm-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 session 7 workshop slide 22 and session 8 tutorial.*

You are reminded that you must ***design your test data so that you always get output for your SQL queries*** - this may require you to add further data as you move through completing the required tasks. Such extra data MUST be added as part of Task 1A (ie. as part of your load of test data). ***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***.

Where a question indicates "Your output must have the form shown below" - this means the same appearance and alignment of columns/data as the sample output shows. Clearly your actual data may be different.

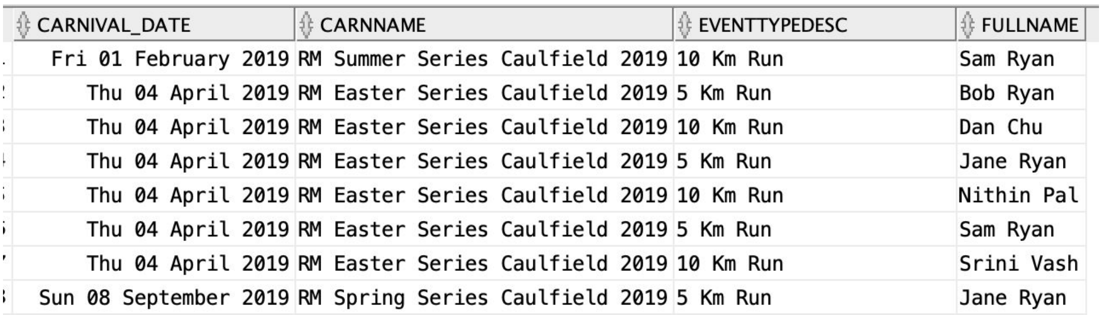
In any query where the sort order (descending or ascending) is not specified, you should use the system default.

(a) List the first name and the last name of the runners who have registered using a Monash University email address (monash.edu) in carnivals organised by Run Monash.

The listing must include

- the date of the carnival(s),
- the name of the carnival(s),
- the event description of the event(s) joined for the carnival(s), and
- the first name and last name of the runners as a single column called fullname.

The listing should be displayed in ascending order of the carnival date and runner's full name within a carnival. Your output must have the form shown below.



CARNIVAL_DATE	CARNNAME	EVENTTYPEDESC	FULLNAME
Fri 01 February 2019	RM Summer Series Caulfield 2019	10 Km Run	Sam Ryan
Thu 04 April 2019	RM Easter Series Caulfield 2019	5 Km Run	Bob Ryan
Thu 04 April 2019	RM Easter Series Caulfield 2019	10 Km Run	Dan Chu
Thu 04 April 2019	RM Easter Series Caulfield 2019	5 Km Run	Jane Ryan
Thu 04 April 2019	RM Easter Series Caulfield 2019	10 Km Run	Nithin Pal
Thu 04 April 2019	RM Easter Series Caulfield 2019	5 Km Run	Sam Ryan
Thu 04 April 2019	RM Easter Series Caulfield 2019	10 Km Run	Srini Vash
Sun 08 September 2019	RM Spring Series Caulfield 2019	5 Km Run	Jane Ryan

[4 marks]

(b) List all registered runners who registered to support a charity as an INDIVIDUAL for the '42.2 km Marathon' event.

The listing must include

- the carnival date,

- the first name and last name of the runner as a single column called runner,
- the charity name,
- the charity contact person, and
- the full description of the supported event in which the runner is running

order the listing in ascending order of carnival date, within a carnival order by the charity name and then by the runners' full name within a supported charity.

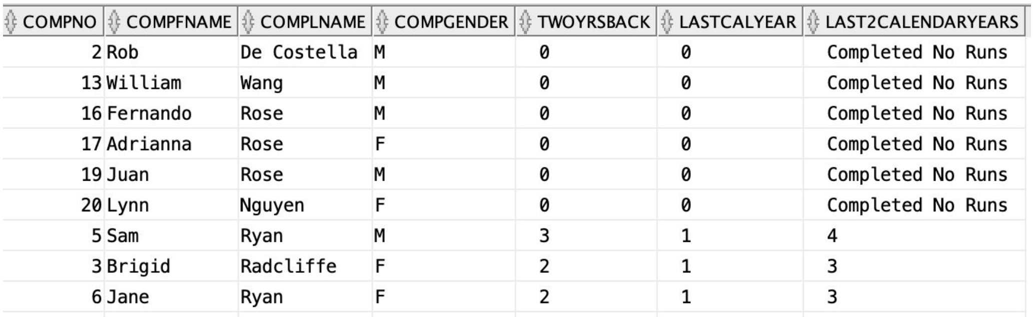
[4 marks]

(c) List the number of events all competitors have completed over the previous two calendar years. For example if this report is run in 2021 it should show the events completed for 2019 and 2020. If it is run in 2022, it should show the events completed for 2021 and 2020, etc. To allow this to occur dates must not be hardcoded as actual values eg. 2019.

The listing must include

- the competitors number,
- the competitors first name,
- the competitors last name,
- the competitors gender,
- how many events they entered two calendar years back,
- how many events they entered for the previous calendar year, and
- how many events they entered across these two previous calendar years. If they have entered no events for the two years, display 'Completed No Runs'

order the listing to show those who have completed no runs first, and for each of the two groups (those who have completed some runs and those who have completed no runs) by competitor number. Your output must have the form shown below.



COMPNO	COMPFNAME	COMPLNAME	COMPGENDER	TWOYRSBACK	LASTCALYEAR	LAST2CALENDARYEARS
2	Rob	De Costella	M	0	0	Completed No Runs
13	William	Wang	M	0	0	Completed No Runs
16	Fernando	Rose	M	0	0	Completed No Runs
17	Adrianna	Rose	F	0	0	Completed No Runs
19	Juan	Rose	M	0	0	Completed No Runs
20	Lynn	Nguyen	F	0	0	Completed No Runs
5	Sam	Ryan	M	3	1	4
3	Brigid	Radcliffe	F	2	1	3
6	Jane	Ryan	F	2	1	3

[5 marks]

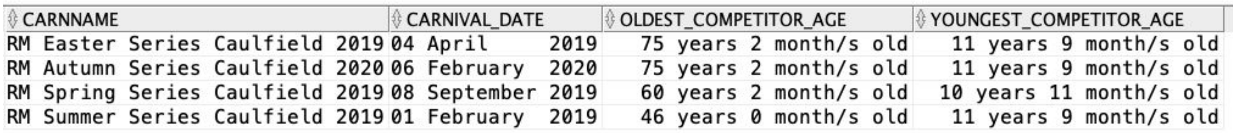
(d) List the oldest runner’s age and youngest runner’s age for each carnival held

The listing must include

- the carnival name,
- the carnival date,
- the oldest runner's age,
- the youngest runner's age

order the output by the oldest runner’s age (oldest first) then by the carnival date.

Your output must have the form:



CARNNAME	CARNIVAL_DATE	OLDEST_COMPETITOR_AGE	YOUNGEST_COMPETITOR_AGE
RM Easter Series Caulfield 2019	04 April 2019	75 years 2 month/s old	11 years 9 month/s old
RM Autumn Series Caulfield 2020	06 February 2020	75 years 2 month/s old	11 years 9 month/s old
RM Spring Series Caulfield 2019	08 September 2019	60 years 2 month/s old	10 years 11 month/s old
RM Summer Series Caulfield 2019	01 February 2019	46 years 0 month/s old	11 years 9 month/s old

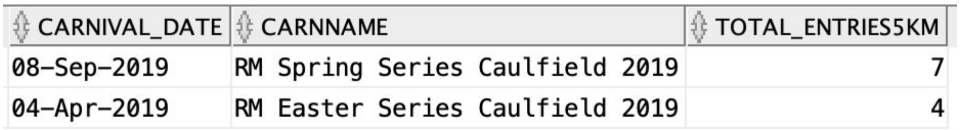
[5 marks]

(e) List the total number of entries/participants in the '5 Km Run'' for each carnival held in the year 2019.

The listing must include

- the date of the carnival,
- the carnival name, and
- the total number of entries/participants in the carnival for this event. Name the column as total_entries5Km

order the listing in descending order of total number of entries, if several carnivals have the same number of entries they should be ordered with the group in ascending carnival date order. Your output must have the form shown below.



CARNIVAL_DATE	CARNNAME	TOTAL_ENTRIES5KM
08-Sep-2019	RM Spring Series Caulfield 2019	7
04-Apr-2019	RM Easter Series Caulfield 2019	4

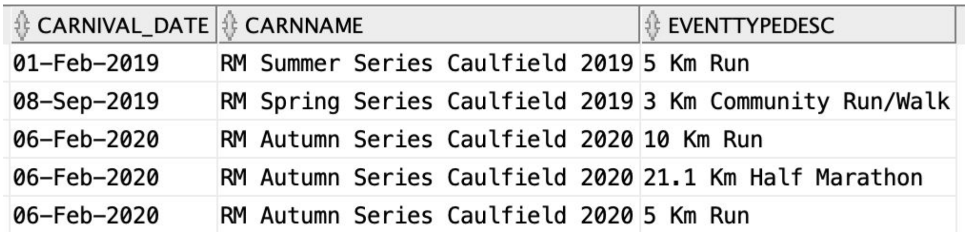
[5 marks]

(f) For all carnivals which have been run by Run Monash (ie. the carnival has been finished), list those events which have had no entries.

The listing must include

- the date of the carnival,
- the carnival name,
- the event description

order the output by event description within the carnival date. Your output must have the form shown below.



CARNIVAL_DATE	CARNNAME	EVENTTYPEDESC
01-Feb-2019	RM Summer Series Caulfield 2019	5 Km Run
08-Sep-2019	RM Spring Series Caulfield 2019	3 Km Community Run/Walk
06-Feb-2020	RM Autumn Series Caulfield 2020	10 Km Run
06-Feb-2020	RM Autumn Series Caulfield 2020	21.1 Km Half Marathon
06-Feb-2020	RM Autumn Series Caulfield 2020	5 Km Run

[6 marks]

(g) For all charities, list the number of times the charity has been nominated (selected for support) by a team, the number of times it has been nominated by an individual and the total number of nominations (team and individual). If no nominations have been made in any of these categories, for a particular charity, the number of nominations must be shown as 0.

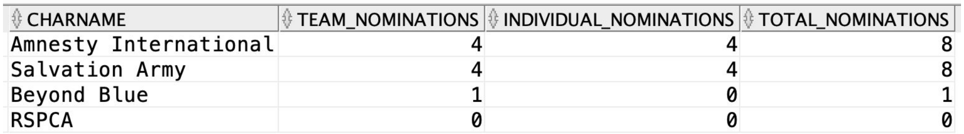
The listing must include

- the charity name,

- the number of times it has been nominated by a team,
- the number of times it has been nominated by an individual, and
- the total number of nominations

order the output by the total number of nominations (highest first), if the total number is the same then order it by the number of team nominations (highest first) then by the number of individual nominations (highest first).

Your output must have the form:



CHARNAME	TEAM_NOMINATIONS	INDIVIDUAL_NOMINATIONS	TOTAL_NOMINATIONS
Amnesty International	4	4	8
Salvation Army	4	4	8
Beyond Blue	1	0	1
RSPCA	0	0	0

[6 marks]

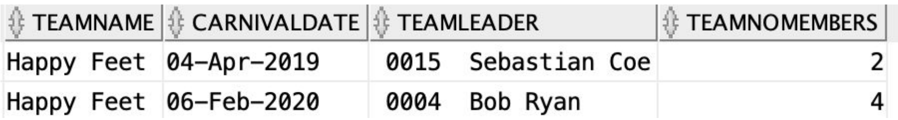
(h) List the team details for each carnival where the most popular team name/s (the team name/s used most often across all carnivals) have been used.

The listing must include

- the team name,
- the date of the carnival,
- the team leaders competitor number as four digits and the first name and last name of the team leader as a single column called TEAMLEADER, and
- the number of members in the team

order the output by the team name and within the team name by carnival date.

Your output must have the form:



TEAMNAME	CARNIVALDATE	TEAMLEADER	TEAMNOMEMBERS
Happy Feet	04-Apr-2019	0015 Sebastian Coe	2
Happy Feet	06-Feb-2020	0004 Bob Ryan	4

[9 marks]

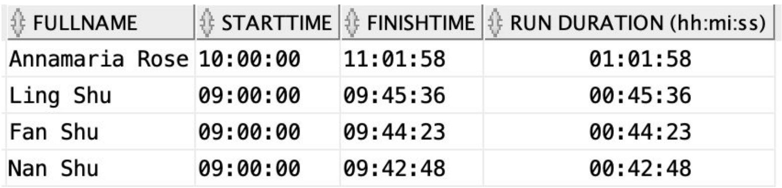
(i) List all the runners who ran in the '5 Km Run' at the RM carnival held on the 8th September 2019 which were slower than the average run (elapsed) time by runners in the '5 Km Run' at the RM carnival held on the 4th April 2019.

The listing must include

- the runner's first name and last name as a single column called fullname,
- the runners start time,
- the runners finish time, and
- the run duration (elapsed time) in hours, minutes and seconds in a single column called 'RUN DURATION (hh:mi:ss)'. The run duration must be shown in the form hh:mi:ss

order the output from the slowest runner to the fastest.

Your output must have the form:



FULLNAME	STARTTIME	FINISHTIME	RUN DURATION (hh:mi:ss)
Annamaria Rose	10:00:00	11:01:58	01:01:58
Ling Shu	09:00:00	09:45:36	00:45:36
Fan Shu	09:00:00	09:44:23	00:44:23
Nan Shu	09:00:00	09:42:48	00:42:48

[11 marks]

TASK 3: Design Modifications (15 marks):

Your answers for these tasks must be placed in the supplied SQL Script **T3-rm-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.

In completing this task, you **must**:

- if you need to add new columns, tables or related constraints, follow the naming conventions used in the data models and schema file which have been provided,
- provide column comments for any new columns that you add, and
- correctly manage any transactions used as part of your solution.

(a) Run Monash organisers have noted that several competitors have enrolled in multiple events in the same carnival which they do not wish to occur.

Change the database structure to prevent a competitor from enrolling in two events for the same carnival.

[2 marks]

(b) Run Monash has decided that they would like to have the elapsed time (finish time - start time) for a runner in a particular event stored as part of the system, rather than having to calculate it every time it is required.

Add a new attribute which will record the runners elapsed time in an event. The time should be stored as the number of minutes elapsed to two decimal places eg. 26.12

This attribute must be initialised to the correct elapsed time based on the data which is currently stored in the system. You should note that the system may contain registrations for future events which currently do not have either a start or finish time.

[3 marks]

(c) Run Monash would also like to store the details of the fastest elapsed time for each event type across all carnival as part of the system.

Add attributes which will record the fastest elapsed time, the carnival name, the competitor's full name (fullname as a single attribute) who holds the record for each event type. The time should be stored as the number of minutes elapsed to two decimal places eg. 26.12.

This attribute must be initialised to the correct fastest elapsed time based on the data which is currently stored in the system. You may use the data stored in the attribute created in Task 3(b) as to initialise these attributes.

[4 marks]

(d) When an emergency contact has been required to be contacted due to a problem with a competitor, there have been several instances of where the listed emergency contact has not been able to be contacted. To help alleviate this issue, Run Monash would like to be able to have a competitor, if they choose to do so, nominate more than one emergency contact.

Change the database structure to allow this to occur.

[6 marks]

TASK 4: PL-SQL (10 marks):

Your answers for these tasks must be placed in the supplied SQL Script **T4-rm-plsql.sql**

These tasks should be attempted only after task 1, task 2 and task 3 have been successfully completed. For each of these questions, as part of your answer, you *must* create a set of SQL commands which will demonstrate the successful operation of your trigger (as an example see the file emp_dept-test-trigger.sql from the emp-dept trigger lab exercise) - *these tests are part of the awarded marks for each question*. Place these commands below your trigger definition for each of the tasks. Ensure your trigger definition finishes with a slash(/) followed by a blank line as detailed in the session 9 workshop and session 10 tutorial. In addition, when coding your trigger(s), you must provide output messages where appropriate.

(a) RM, from now on, would like to automatically maintain the number of team members (TeamNoMembers). When an entry data is inserted/updated/deleted, the number of team members must be updated accordingly. RM would also like to ensure that the maximum members for a team is 5. If the number of team members has reached 5, then the trigger should prevent the action of adding new members to the team. Code a *single* trigger to enforce both of these requirements.

[4 marks]

(b) RM, would like to automatically calculate, and store, the elapsed time when the finish time for an entry is added after the competitor has finished their run. If the new elapsed time is the fastest elapsed time for a particular event type, then the event type's fastest elapsed time, the carnival name and the competitor's full name who holds the record must also be updated. You may assume that the finish time will only be added once for a given entry. Code a single trigger to enforce these requirements.

[6 marks]

SUBMISSION REQUIREMENTS

Due Date: **Friday, 12th February 2021 5PM AEDT**

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 five files you are **required** to submit:

- T1a-rm-insert.sql
- T1b-rm-dm.sql
- T2-rm-queries.sql
- T3-rm-mods.sql
- T4-rm-plsql.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.

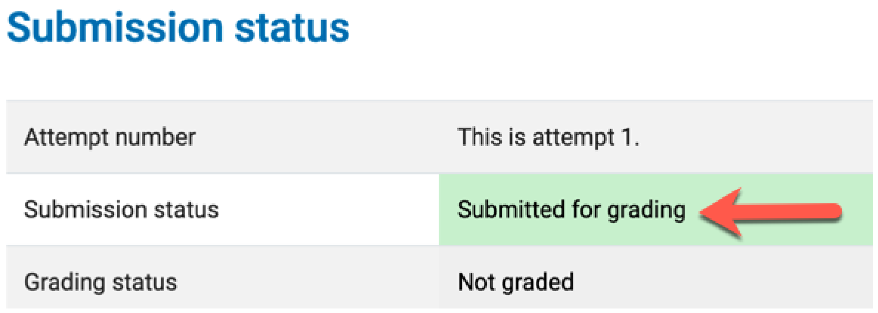
Do not zip these files into one zip archive, submit five independent SQL scripts. The individual files must also have been pushed to the FIT GitLab server with an appropriate history as you developed your solutions.

Late submission will incur penalties at the rate of -5 marks for every 12 hours the submission is late.

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 include slash (/) followed by a blank line after the create trigger statements,
- do not provide output messages where appropriate when coding triggers,
- do not have an appropriate development history on the FIT GitLab server for all source files (at least four pushes required).