# Exam 2017, questions and answers

Databases (Monash University)

# Monash University

## FIT2094

## SAMPLE EXAM PAPER

**PART A. Multiple Choice Questions. (10 x 2 marks = 20 marks).**

**Mark your selection by placing a √ or a X through your selected answer for example ☑ or ☒**
**If you change your selection during the review of your paper, prior to the end of the Examination, make sure that the alteration is clear.**

**Note: There is only ONE correct answer for each question. There is no penalty for incorrect answer.**

1. Assume you are using the MINUS operator to combine the results from two tables with identical structure, CUSTOMER and CUSTOMER_2. The CUSTOMER table contains 10 rows, while the CUSTOMER_2 table contains 7 rows. Customers Dunne and Olowski are included in the CUSTOMER table as well as in the CUSTOMER_2 table (and no other customers are in both tables). How many records are returned when using the MINUS operator - i.e., in analyzing CUSTOMER MINUS CUSTOMER_2?
   a. 0
   b. 2
   **c. 8**
   d. 10

2. A ____ relationship exists when an association is maintained within a single entity.

   **a. unary**
   b. ternary
   c. binary
   d. weak

3. Which statement is correct with respect to the RELATIONAL MODEL properties?

   a. **The order of attributes is immaterial**.
   b. Duplicate rows are possible.
   c. Attribute may have multiple values.
   d. Tuples should be addressed by their position in the relation.

4. Which statement is correct in relation to a VIEW?

   a. The data in a view can always be updated.
   b. **The data in a view is generated only when the view is used.**
   c. The data in view can always be inserted using INSERT statement.
   d. A derived column is not allowed in a view.

5. A table that is in 2NF and contains no transitive dependencies is said to be in ____.

    a. 1NF
    b. 2NF
    c. **3NF**
    d. 4NF

6. The following SQL is executed in ORACLE and produces an error message. What is the most likely cause of the error message?

```
SELECT student_id, student_name
FROM student
WHERE course='MAIT'
UNION
SELECT student_id, student_name, student_dob
FROM student
WHERE course='MIT'
```

    a. The union operation is not supported for two tables.
    b. The attribute student_name does not have the correct data type.
    c. The union operation should not be used for retrieving data from a single table.
    d. **The two select statements are not union compatible**.

7. ANSI SPARC recommends view integration into database design. The recommendation suggests three levels of design. What are the levels?

    a. External, conceptual, database
    b. Conceptual, logical, physical
    c. External, conceptual, logical
    d. **External, conceptual, internal**.

8. An ER diagram needs to be developed to represent the fact that an employee may supervise another employee. Both the supervisor and the supervisee are employees. What would be the best way for you to model this fact?

    a. Supervisor is a strong entity and the supervisee is a weak entity.
    b. **A recursive relationship called supervises on an entity called employee**.
    c. A multi-varied attribute called supervisor in an entity called employee.
    d. A many to many relationship between an entity called supervisor and an entity called supervisee.

9. A trigger to perform update cascade needs to be implemented to ensure data integrity between EMPLOYEE and DEPARTMENT on the deptno. An update of deptno in the DEPARTMENT should cause all the associated tuples in the EMPLOYEE table be updated. Which statement is correct in relation to this trigger?

    a. The trigger should be created for the EMPLOYEE table.
    b. It is not possible to implement this trigger because it will involve a mutating table.
    c. **The trigger can be implemented as row level trigger**.
    d. The trigger has to contain an IF.. THEN.. ELSE..statement.

10. Which of the following is an example of a soft crash in database?

    a. Disk crash.
    b. **Power Failure**.
    c. Software bugs.
    d. Concurrency.

**END OF PART A**

## PART B. Answer all questions on the space provided. Total: 80 marks

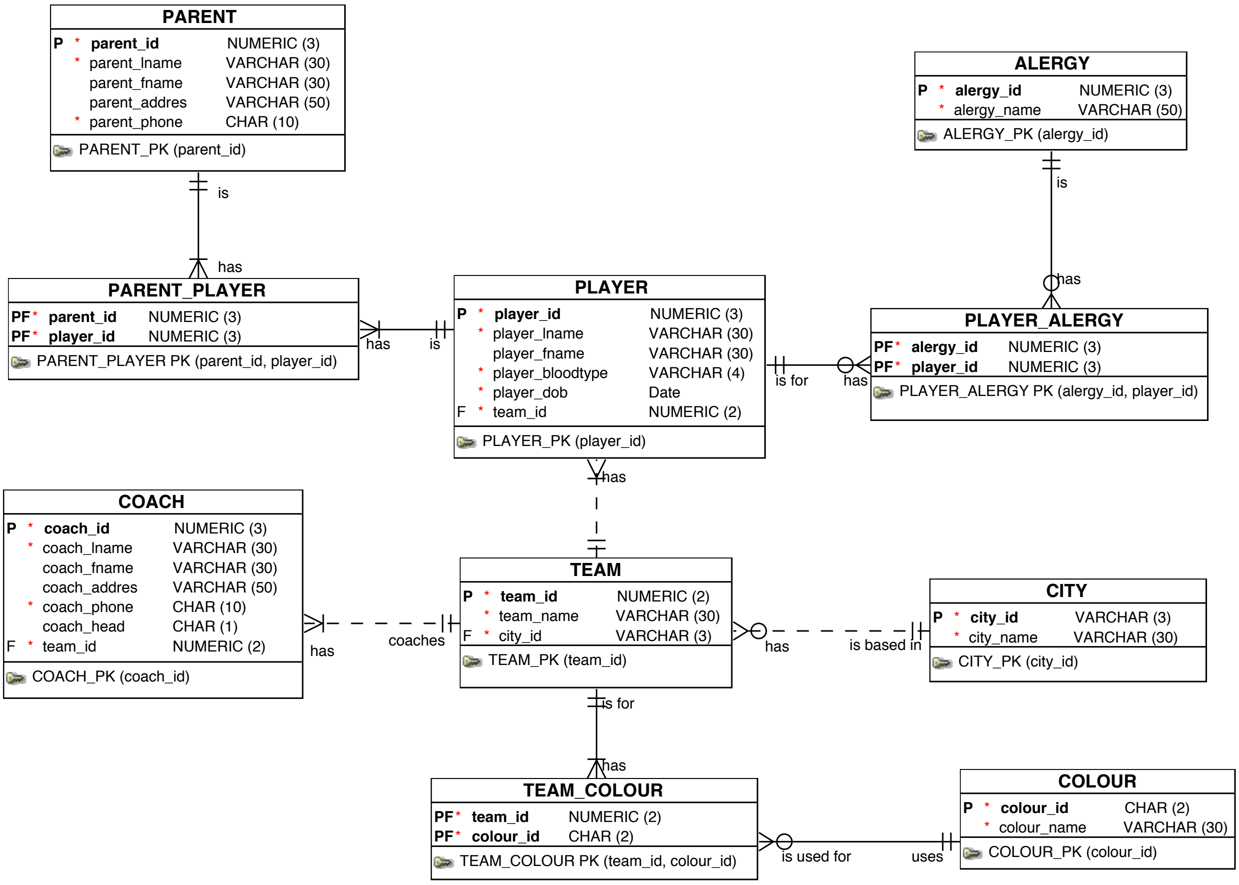
1. Draw a *logical level model* using *crowsfoot notations* that depicts the situation described below. (**15 marks**)

The local under-sixteen football league needs a database to help track teams, children that sign up to play in the league, the parents of these children and the coaches for each team.

The league wishes to record the details for each parent of a particular player (the parents last name, first name, phone contact number and address).

For each player the system needs to record the player's last name, first name, blood type and their date of birth. Any allergies that the player has also need to be recorded. A player may only play in one team with this league.

Each team is given a unique id, the system needs to record this id, the teams name and the city their home ground is located in (a city may have several under-sixteen teams based in it). A team's colours are also to be recorded, teams may choose to use a single or multiple team colours. A team may have several coaches – one of the coaches is designated as the head coach. A coach is only permitted to coach one team in the under-sixteen league. All communications from the league to the team are via the head coach. To be registered to play in this league a team must have a head coach and at least one player. The database needs to track a coach's first and last name, phone and address contact details and the team that they are coaching.



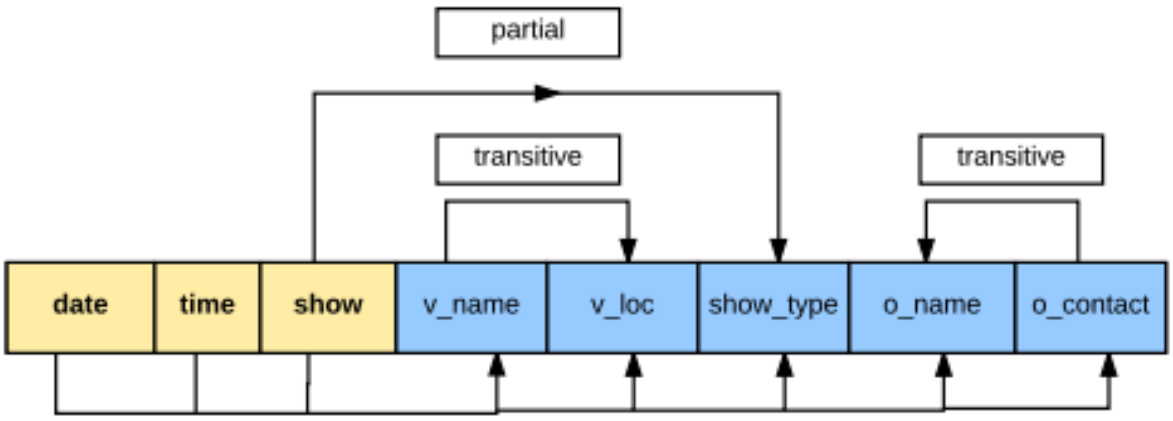
2. Normalisation (**10 marks**)

   Monash University owns several performance halls that are used by organizations within and outside Monash University. The following table shows the booking information for several performances across different venues in Monash University. Multiple performances or shows can be organized at the same time across multiple venues. The following situations are observed during the operation of the performance halls:

   - Each show is organised by an organisation.
   - A single contact number is kept for an organiser.
   - A show can be scheduled on multiple days, for example Melbourne Symphony-Summer Classic has two performances on the 12-Jan-2015 and 17-Jan-2015.
   - A show can be scheduled twice a day (matinee and night).

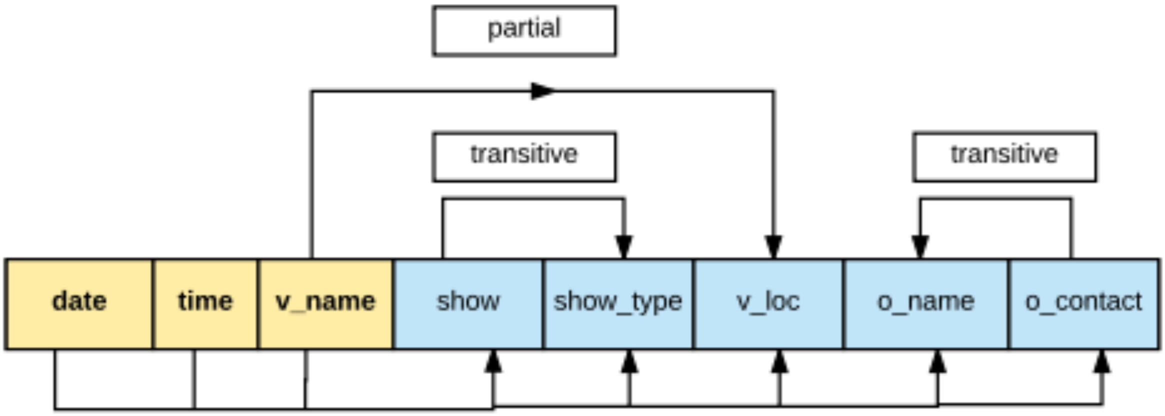
| **Start Time** | **Date** | **Venue** | **Venue Location** | **Show Type** | **Show** | **Organiser** | **Organiser's contact** |
|---|---|---|---|---|---|---|---|
| 8 PM | 12-Jan-2015 | Robert Blackwood Hall | Clayton | Music concert | Melbourne Symphony – Summer Classic | MSO | (03) 99021212 |
| 8 PM | 12-Jan-2015 | K3.24 | Caulfield | Comedy | Adam Hill | Melbourne Comedy Festival | (03) 99031456 |
| 2 PM | 14-Jan-2015 | Robert Blackwood Hall | Clayton | Musical | Cats | Monash Student Association | (03) 99012233 |
| 8 PM | 14-Jan-2015 | Alexander Theatre | Clayton | Comedy | Dave Hughes | Melbourne Comedy Festival | (03) 99031456 |
| 8 PM | 16-Jan-2015 | Robert Blackwood Hall | Clayton | Music concert | Hoodoo Guru | Mushroom Promoter | (02) 90021002 |
| 8 PM | 17-Jan-2015 | Robert Blackwood Hall | Clayton | Music concert | Melbourne Symphony – Summer Classic | MSO | (03) 99021212 |

a) Draw a dependency diagram for this table.
**date, start_time, show,** v_name, v_location, show_type, o_name, o_contact



=========== ALTERNATIVE ======================

**date, start_time, v_name,** show, v_location, show_type, o_name, o_contact



b) Convert the table shown above to Third Normal Form (3NF), showing each stage of the process. Clearly state any assumptions that you make. You should not add new attribute.

1NF
**PERFORMANCE(date, start_time, show,** v_name, v_location, show_type, o_name, o_contact)

**2NF**

**PERFORMANCE(date, start_time, show,** v_name, v_location, o_name, o_contact)

**EVENT(show,** show_type)

**3NF**

**PERFORMANCE (date, start_time, show,** v_name, o_contact)

**VENUE(v_name,** v_location)

**EVENT(show,** show_type)

**ORGANISER(o_contact**, o_name)

=========== ALTERNATIVE =====================

1NF

**BOOKING(date, start_time, v_name,** show, v_location, show_type, o_name, o_contact)

2NF

**BOOKING(date, start_time, v_name,** show, show_type, o_name, o_contact)

**VENUE(v_name,** v_location)

3NF

**BOOKING(date, start_time, v_name,** show, o_contact)

**VENUE(v_name,** v_location)

**EVENT(show,** show_type)

**ORGANISER(o_contact**, o_name)

3. a. What does is meant by the ACID properties in a database transaction? **(5 marks)**

A = atomicity
A transaction should be processed entirely, transaction can be either committed or rollback.

C = consistency
A transaction should bring a database from one consistent state to another consistent state.

I = Isolation
A result of a transaction should not be seen by another transaction until it has been commited.

D = Durability
Once a transaction is committed, the result of this transaction should never be lost.

b. Give an example of how adherence to ACID properties can prevent the reading "uncommitted data" problem. **(5 marks)**

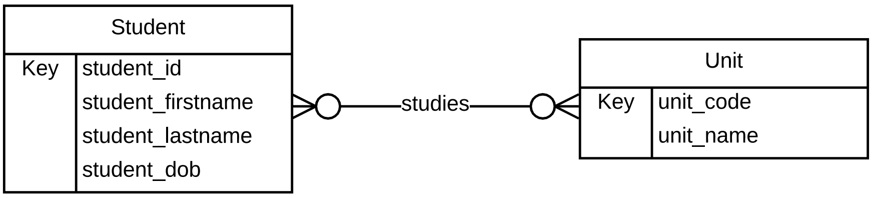
Reading "uncommitted data" occurs when on transaction (transaction A) is allowed to read some data that has been updated by another transaction (transaction B) but is rolled back due to errors. The data that has been read by transaction A will not reflect the fact that the update made by transaction B should not be considered to be finished as the commit state was never reached by transaction B. For example, assume QOH = 20, transaction B updated QOH=QOH+20 (QOH=40), before its reaches the COMMIT statement, transaction A reads the QOH which has the value of 40. After transaction A read the QOH, transaction B rolled back due to error and put the QOH value back to 20. The QOH reads by A is incorrect, it should be 20 rather than 40.

If a protocol follows an ACID principle is devised by DBMS using a locking mechanism, DBMS will ensure that transaction A can only read QOH once transaction B has reached a commit stage. It is achieved by granting the exclusive lock on QOH to transaction B when transaction B reads QOH in order to update the value. The DBMS should deny the request for any type (share or exclusive) lock by transaction A when it reads the QOH.

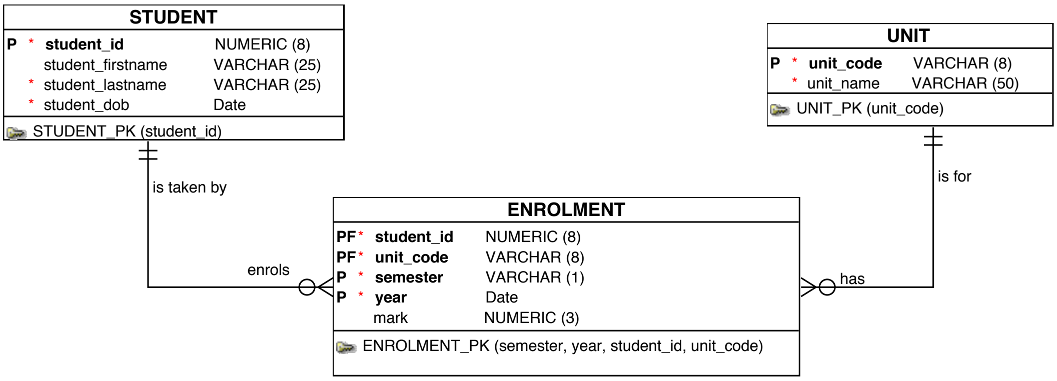
The atomicity and isolation will ensure that transaction A will be able to read (request for shared lock is granted) QOH ONLY when transaction B commits the changes made to QOH or has been rolled back (completed or never start - atomicity). Hence, when transaction B was rolled back, transaction A should never see the value of QOH = 40. It should read QOH=20 after the transaction B was rolled back.

Once a transaction is committed or rolled back, the result of the transaction should never be lost according to the durability. Hence, even in a situation of power failure during the execution of transaction B and the transaction B was rolled back, the restart process will ensure QOH will be 20 (the value before transaction B started) because it is the last consistent value in the database before transaction B started.

6. An entity relationship diagram to depict a relationship between STUDENT and UNIT is given below



(i) Map the above conceptual model to a logical level diagram relational model using Crow's foot notation. Include in this model the attributes ***semester, year*** and ***mark***. The mark is allocated for a student for a given unit in a given semester and year. **(5 marks)**


STUDENT
P * student_id NUMERIC (8)
student_firstname VARCHAR (25)
* student_lastname VARCHAR (25)
* student_dob Date
STUDENT_PK (student_id)
UNIT
P * unit_code VARCHAR (8)
* unit_name VARCHAR (50)
UNIT_PK (unit_code)
is taken by
is for
enrols
has
ENROLMENT
PF * student_id NUMERIC (8)
PF * unit_code VARCHAR (8)
P * semester VARCHAR (1)
P * year Date
mark NUMERIC (3)
ENROLMENT_PK (semester, year, student_id, unit_code)

(ii) Write the create table statement for **the table that contains the semester, year and mark attributes**. Include in the create table statement a constraint that ensures that the mark is limited to the range of 0 to 100 inclusive. You DO NOT need to write create table statements for the STUDENT and the UNIT tables. In writing the create table statement, you can assume that create table statements for the STUDENT and the UNIT table have been included in the schema and will be executed prior to the execution of the create table statement that you have to write. **(5 marks)**

```
CREATE TABLE ENROLMENT (
     student_id NUMBER (8)  NOT NULL ,
     unit_code VARCHAR2 (8)  NOT NULL ,
     semester VARCHAR2 (1)  NOT NULL ,
     year DATE  NOT NULL ,
     mark NUMBER (3) CHECK ( mark BETWEEN 0 and 100)
     CONSTRAINT ENROLMENT_PK PRIMARY KEY ( semester,
year, student_id, unit_code ) ;


ALTER TABLE ENROLMENT
    ADD CONSTRAINT student_enrolment_fk FOREIGN KEY
    ( student_id)
    REFERENCES STUDENT
    ( student_id) ;


ALTER TABLE ENROLMENT
    ADD CONSTRAINT unit_enrolment_fk FOREIGN KEY
    ( unit_code )
    REFERENCES UNIT
    ( unit_code) ;
```