SAMPLE EXAM SOLUTIONS ONLY. DO NOT DISTRIBUTE UNTIL WEEK 13.

Monash University

FIT2094

SAMPLE EXAM PAPER S1 2019

- Page 1 is normally reserved for the Exams Coversheet.

- IMPORTANT NOTE: This Sample Exam serves to provide a general overview of the general structure of the exam paper only.

- To protect the integrity of the exam:
NO ACTUAL EXAM QUESTIONS are included; and the COMPOSITION OF THE SUBQUESTIONS are SUBJECT TO CHANGE.

- Students are reminded that all content specified by the Unit Guide is examinable, including but not limited to Pre-reading (weekly Coronel & Morris chapters) + Lecture Notes + Tute Notes + all other Moodle Material (except where explicitly stated).

Note before beginning:

You are required to write all answers in the provided script book.
ANYTHING WRITTEN IN THIS EXAM QUESTION BOOKLET IS NOT MARKED.

(Best wishes and good luck, from all of us at Unit Management).

Q1 The Relational Model (Total=10 marks)

(1a) ATTRIBUTES

A company wishes to record the following attributes about their employees: employees ID, department number, name, home address, education qualifications and skills which the employee has. A small sample of data is show below:

Employee ID	Department Number	Employee Name	Home Address	Qualification	Skill
101	21	Given name: Joe Family name: Bloggs	Street: 12 Wide Rd Town: Mytown Postcode: 1234	-Bachelor of Commerce -MBA	-Project Management -Hadoop -R
102	13	Given name: Wendy Family name: Xiu	Street: 55 Narrow St Town: Mytown Postcode: 1234	-Bachelor of Computer Science -Master of IT Doctor of Philosophy	-SQL -PL/SQL
103	13	Given name: Sarah Family name: Green	Street: 25 High St Rd Town: Mytown Postcode: 1234	-Certificate IV in Business Administration	-SQL -Java -Python

Using ONLY the data from the sample above, explain the difference between a multivalued attribute and a composite attribute. Justify your answer with CORRECT examples.

(1b) RELATIONAL ALGEBRA

International conservation programmes include cooperation between zoos around the world to maintain and breed populations of animals, especially species that may become threatened in the wild. In this programme, different zoos share data about animals and about the health problems they encounter in keeping them.

The database contains basic data about zoos, about the species represented, about individual animals, and about the veterinary surgeons who tend to their health. A record is kept of ailments suffered by each animal and how they were treated.

```
    Taxon
            taxon_id      char(10)          NN     PK
            genus         varchar2(30)      NN
            species       varchar2(30)      NN
            description   varchar2(1000)
            ref_url       char(17)
    Zoo
            zoo_name      varchar2(30)      NN     PK
            country       varchar2(30)      NN
            city          varchar2(30)      NN
            zoo_phone     number(12)
            zoo_address   varchar2(30)      NN
    Animal
            animal_id     number(4)         NN     PK
            taxon_id      char(10)          NN     FK
            zoo_name      varchar2(30)      NN     FK
            dob           date
            sex           char(1)           NN
    Vet
            vet_name      varchar2(30)      NN     PK
            vet_phone     number(12)               NN
            vet_address   varchar2(30)      NN
            vet_country   varchar2(30)      NN
            vet_city      varchar2(30)      NN
    Ailment
            case_date     date        NN    PK
            animal_id     number(4)         NN     PFK
            vet_name      varchar2(30)      NN     FK
            condition     varchar2(30)      NN
            description   varchar2(1000)

    Please Note:  NN = Not Null, PK = Primary key, FK = Foreign key
```

*** CONTINUED NEXT PAGE ***

With reference to the description of the Zoo animal care database above, write Relational Algebra statements, using symbolic notation, to produce the following information.

These rules MUST be followed when writing your algebra:
- you MUST use symbolic notation,
- if the relational algebra is complex and contains several operations, you may wish to split your answer into several steps, such as:
  R1 <- relational algebra expression-1
  R2 <- relational algebra expression-2 that uses R1 as an input, etc,
- and, where a query has several possible solutions, your relational algebra must represent an EFFICIENT solution.

(i) List the names and phone numbers of all vets **[2 marks]**
(ii) List the animal id and zoo name for all female 'F" animals in the system **[2 marks]**
(iii) List the genus and species of all animals from South African zoos that have received treatment. **[2 marks]**

*** Q1 ENDS HERE ***

`(1a) SAMPLE ANSWERS`

`A Multivalued attribute is an attribute which has several possible values for a given instance — Skills. For example, employee 102 has skills in PL/SQL and SQL`

`A Composite attribute is an attribute which has a single value for a given instance which may be subdivided into other values - Employee name or Home Address. For example, employee 101's name can be subdivided into given and family`

`(1b) SAMPLE ANSWERS`

(i) a1 ← π vet_name, vet_phone (VET)

(ii) a2 ← π animal_id, zoo_name (s sex = "F" (ANIMAL))

(iii) a3a ← π zoo_name (σ country = "South Africa" (ZOO))
a3b ← π taxon_id, animal_id (a3a ⋈zoo_name ANIMAL)
a3c ← π taxon_id (a3b ⋈animal_id AILMENT)
a3d ← π genus, species (a3c ⋈taxon_id TAXON)
OR

A3 ← π genus, species (TAXON ⋈taxon_id
              (π taxon_id (AILMENT ⋈animal_id
                (π taxon_id, animal_id (ANIMAL ⋈zoo_name
                      (π zoo_name (σ country = "South Africa" (ZOO)))
)
)

)

)

Q2. Database Design (Total=20 marks)

Case study below needs to be understood before commencing.

The Charity Football League - a fundraising under-sixteen football league for local communities, whose format is modelled after the English Premier League - needs a database to help track teams, children that sign up to play in the league, the parents of these football players 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 team's name and the city their home ground is located in (a city may have several under-sixteen teams based in it - e.g. Manchester can have Manchester Victory or Manchester Rovers).

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.

Create **a logical level diagram using Crow's foot notations** to represent the "Charity Football League" data requirements described above. Clearly state any assumptions you make when creating the model. Be sure to include all attributes. Identify clearly the Primary Keys and Foreign Keys, as part of your design. You do not need to include the data type of the attributes.

*** Q2 ENDS HERE ***

(2) SAMPLE ANSWERS

NB: Correction - the word 'ALERGY' is accidentally misspelt. It should be 'ALLERGY'. However, it should bear no effect on the actual design.
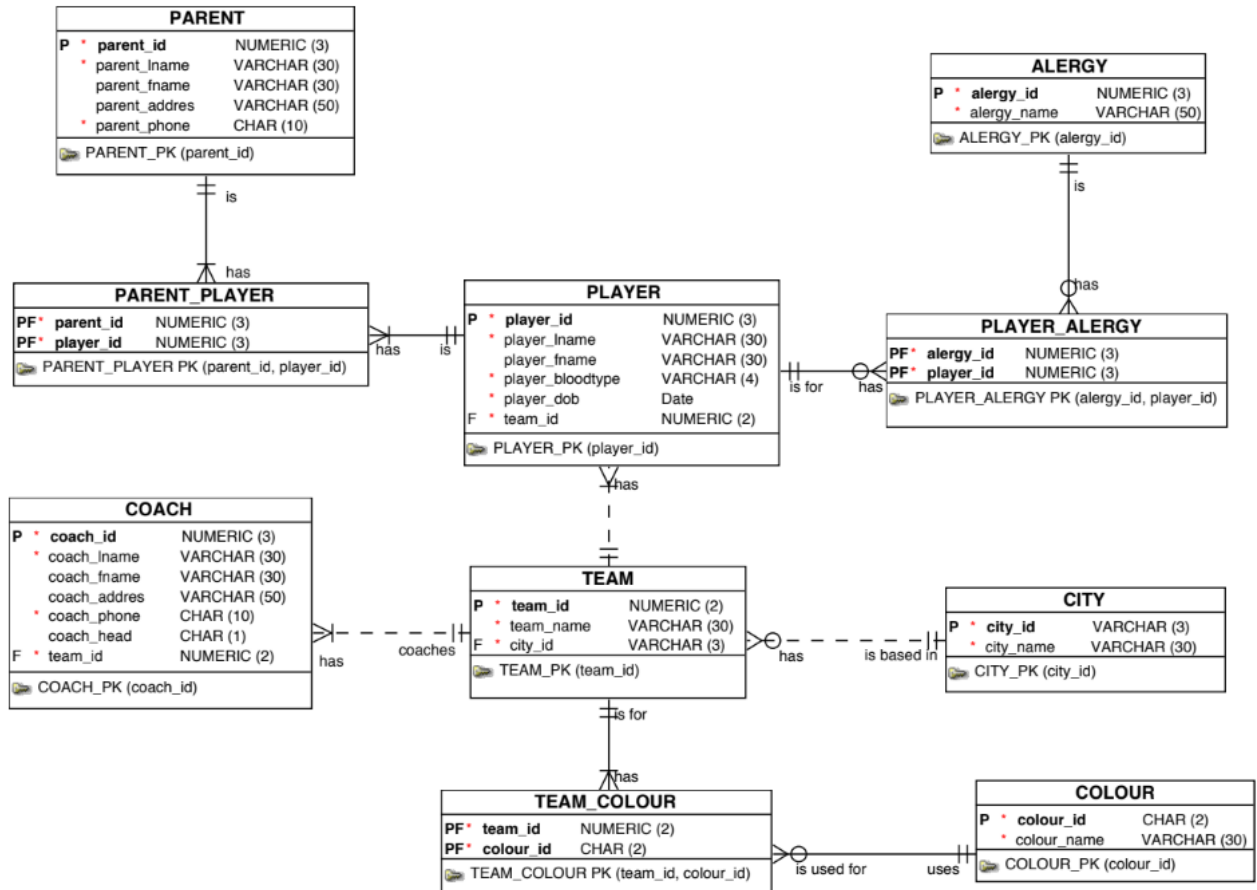
Q3. Normalisation (Total=20 marks).

Refer to the following business case and Report.

Monash University owns several performance halls that are used by organizations within and outside Monash University.

The **following report** 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. **For consistency, we call performances/shows as events.**

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-2019 and 17-Jan-2019.
- 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-2019	Robert Blackwood Hall	Clayton	Music concert	Melbourne Symphony – Summer Classic	MSO	(03) 99021212
8 PM	12-Jan-2019	K3.24	Caulfield	Comedy	Adam Hill	Melbourne Comedy Festival	(03) 99031456
2 PM	14-Jan-2019	Robert Blackwood Hall	Clayton	Musical	Cats	Monash Student Association	(03) 99012233
8 PM	14-Jan-2019	Alexander Theatre	Clayton	Comedy	Dave Hughes	Melbourne Comedy Festival	(03) 99031456
8 PM	16-Jan-2019	Robert Blackwood Hall	Clayton	Music concert	Hoodoo Gurus	Mushroom Promoter	(02) 90021002
8 PM	17-Jan-2019	Robert Blackwood Hall	Clayton	Music concert	Melbourne Symphony – Summer Classic	MSO	(03) 99021212

From UNF, convert the table shown above to Third Normal Form (3NF), showing each stage of the process. Clearly state any assumptions that you make. For the report only, show ALL

dependencies via required dependency diagrams. All attributes must be included, no surrogate keys may be added during the normalisation.
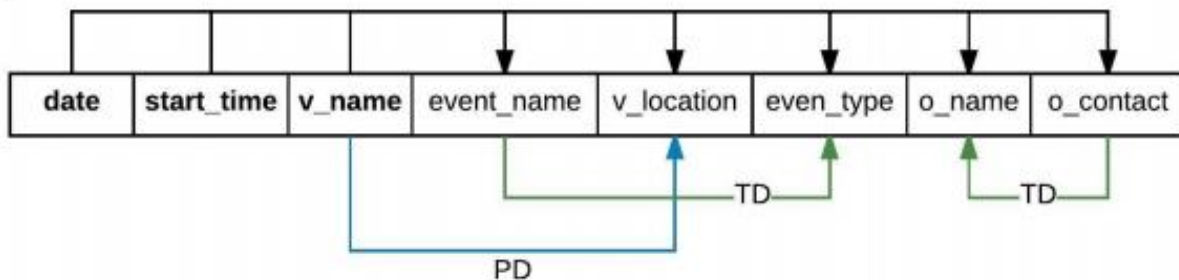
*** Q3 ENDS HERE ***
(3) SAMPLE ANSWERS

DEPENDENCY DIAGRAM FOR REPORT ONLY:

Simple:
date, start_time, v_name, event_name, v_location, event_type, o_name, o_contact

General:
date, start_time, v_name -> event_name, o_contact FULL
v_name -> v_location PARTIAL based on the PK date, start_time, v_name
event_name -> event_type PARTIAL based on the candidate key date, start-time, event_name
o_contact -> o_name TRANSITIVE

UNF same as 1NF. No repeating groups.

1NF
BOOKING (date, start_time, v_name, event_name, v_location, event_type, o_name, o_contact)

2NF
BOOKING (date, start_time, v_name, event_name, o_name, o_contact)
VENUE (v_name, v_location)
EVENT (event_name, event_type)

3NF
BOOKING (date, start_time, v_name, event_name, o_contact)
VENUE (v_name, v_location)
EVENT (event_name, event_type)
ORGANISER (o_contact, o_name)

Q4. Transaction Management (Total=10 marks)

(4a)

Using an example, illustrate and explain what the lost update problem is where two concurrent transactions are updating the same data element. Then, define the term ATOMICITY and explain how it can mitigate the problem above. **[7 marks]**

(4b)

Given the following hypothetical scenario where multiple users are accessing a slow database each with their own transactions.

Assume T1 = user anne; T2 = user bruce; T3 = user charles; T4 = user david
- anne acquires a shared lock for record A then reads it.
- bruce acquires a shared lock for record B then reads it.
- charles wants to write to record A, but anne has not finished.
- david acquires an exclusive lock for record C then changes it.
- emma acquires a shared lock for record A then reads it.
- david wants to write to record A, but anne has not finished.

Illustrate the scenario above using a Wait-For Graph (WFG), with the correct notation (T1, T2... etc) instead of usernames.

Then, explain the presence/absence of any deadlock.
You must justify your answer based on the WFG. **[3 marks]**

*** Q4 ENDS HERE ***

(4a) SAMPLE ANSWERS

A good sample answer is Coronel & Morris Table 10.3

TABLE 10.3	Lost Updates		
TIME	**TRANSACTION**	**STEP**	**STORED VALUE**
1	T1	Read PROD_QOH	35
2	T2	Read PROD_QOH	35
3	T1	PROD_QOH = 35 + 100	
4	T2	PROD_QOH = 35 − 30	
5	T1	Write PROD_QOH (Lost update)	135
6	T2	Write PROD_QOH	5

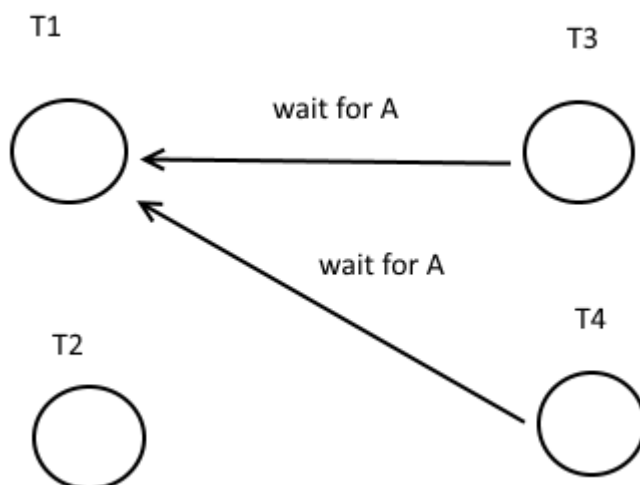**Definition of ATOMICITY per lecture notes**
"all database operations (SQL requests) of a transaction must be entirely completed or entirely aborted"

Application to Table 10.3
"All database operations of T1" must be entirely completed before T2.

(4b) SAMPLE ANSWERS
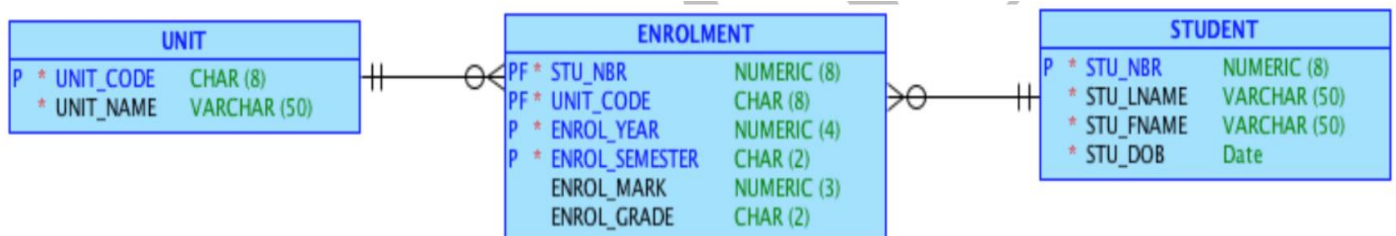**WFG - must be using Tn notation - not user names.**

T3 waiting for T1 (charles waiting for anne). T4 waiting for T1 (david waiting for anne).
However there is NO CYCLE = NO DEADLOCK.

Q5. FIT2094: Intermediate SQL, Database Theory and Implementation (Total = 40 marks)

For questions 5(a), 5(b), 5(c) refer to the ENROLMENT data model below, modified from case studies found in the unit.

(5a) SQL (DQL) SELECT [10 marks]

For the ENROLMENT data model, construct SQL SELECT statements for the following.

i. Select all unique student names for all the students who are enrolled in FIT units (prefix of unitcode is exactly 'FIT') as well as BIZ units (prefix of unitcode is 'BIZ'). You CANNOT use any set operations.

ii. Select the unit code, unit name - and for each unit, the total number of enrolments across all semesters. Use a column alias to describe the last column more sensibly, as TOTAL_ENROLMENTS. Sorting must be done appropriately, according to the following example of the expected result format:

```
UNIT_CODE    UNIT_NAME              TOTAL_ENROLMENTS
ACC1001      Accounting Basics      20
ACC2002      Technical Accounting   117
ACC6006      MYOB Accounts          63
etc...
```

iii. For each unit, find the students who obtained the lowest mark in the unit. You MUST use the 'Derived Table' (subquery inline) style and CANNOT use Views.

```
(5a) SAMPLE ANSWERS:

-- Part i
SELECT DISTINCT stu_lname, stu_fname
FROM enrolment e JOIN student s ON e.stu_nbr = s.stu_nbr
WHERE e.unit_code LIKE 'FIT%' OR e.unit_code LIKE 'BIZ%';

-- Part ii
SELECT unit_code, count(*) as TOTAL_ENROLMENTS
FROM enrolment e
GROUP BY unit_code
ORDER BY unit_code ASC;


-- Part iii
-- refer Lecture 9 --
SELECT e.stu_nbr, e.unit_code, e.enrol_mark
FROM enrolment e
JOIN (SELECT unit_code, max(enrol_mark) as max_mark
      FROM enrolment
      GROUP BY unit_code) max_table
ON e.unit_code = max_table.unit_code
AND e.enrol_mark = max_table.max_mark
ORDER BY unit_code, stu_nbr;
```

(5b) SQL DML [10 marks]

Refer to the ENROLMENT data model, and assume ALL the tables have ALL been created correctly in Oracle. Also assume the order of attributes the programmer used in CREATE TABLE statements follow the order it appears in the data model, top-to-bottom (e.g. for UNIT, UNIT_CODE is first, then UNIT_NAME).

i. Add an enrolment with the following details:
- student whose ID needs to be looked up, with DOB = 25 Dec 2000 and surname Smith;
- unit whose code needs to be looked up, with the exact unit name 'Graphics';
- first semester; and
- year needs to be extracted from today's date and converted to characters appropriately.
- no grades nor marks as of now.

ii. The script shown below is used to add 2 students and their enrolments into two tables STUDENT and ENROLMENT. The new students are "James Bond" and "Bruce Lee". James Bond wants to enrol into FIT1004 and FIT1001, whereas Bruce Lee wants to enrol into FIT1004.

```
-- Start of INSERT script
INSERT INTO student VALUES (sno_seq.nextval,'Bond','James',to_date('01-Jan-
1994','dd-mon-yyyy'));

INSERT INTO student VALUES (sno_seq.nextval,'Lee','Bruce',to_date('01-Feb-
1994','dd-mon-yyyy'));

INSERT INTO enrolment VALUES (sno_seq.currval,'FIT1004',2012,1,0,'NA');
INSERT INTO enrolment VALUES (sno_seq.currval,'FIT1001',2012,1,0,'NA');
INSERT INTO enrolment VALUES (sno_seq.currval,'FIT1004',2012,1,0,'NA');
COMMIT;
-- Finish of INSERT script
```

An Oracle sequence called sno_seq has been created for auto-generating of the student number in the database. The units listed in the script (e.g., FIT1004, FIT1001) exist in the UNIT table.

What problems will be associated with the execution of the above script?

(5b) SAMPLE ANSWERS:


```
-- Part i
INSERT INTO enrolment VALUES (
    (SELECT stu_nbr FROM student
     WHERE stu_dob = to_date('25-Dec-2000','dd-Mon-yyyy')
     AND stu_lname = 'Smith'),
    (SELECT unit_code FROM unit
     WHERE unit_name = 'Graphics'),
    to_char(sysdate,'yyyy'),
    '1',
    NULL,
    NULL
);
```


```
-- Part ii
An error message will be generated when the fifth INSERT statement
is executed by the DBMS. The error will state there is a duplicate
PK identified. Bruce Lee will be enrolled into FIT1004 twice for
S1, 2015 offering. James Bond will not be enrolled in FIT1004.
Bruce Lee will be enrolled in both FIT1004 and FIT1001.
```

(5c) INTERMEDIATE DDL / Triggers [10 marks]

i. Refer to the ENROLMENTS data model,.
Write the create table statement for the ENROLMENT table. Include in the create table statement a constraint that ensures that the mark is limited to the range of 0 to 100 inclusive. Include also a constraint that ensures the grade is either N, P, C, D, HD, or NULL.

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.

ii. Lets say you are removing a table called TEST2 from a test database.
You issue the following SQL command verbatim:
DROP TABLE TEST2 CASCADE CONSTRAINTS PURGE;

What is the reason behind the usage of both PURGE and CASCADE CONSTRAINTS?

iii. Give a brief definition of Oracle's PL/SQL Triggers.
Provide a use case/example on how Triggers can be used for auditing.

(5c) SAMPLE ANSWERS:

```
-- Part i
CREATE TABLE ENROLMENT (
    stu_nbr NUMBER(8) NOT NULL,
    unit_code CHAR(8) NOT NULL,
    enrol_year NUMBER(4) NOT NULL,
    enrol_semester CHAR(2) NOT NULL,
    enrol_mark NUMBER(3),
    enrol_grade CHAR(2),

    CHECK (enrol_mark BETWEEN 0 and 100),
    CHECK (enrol_grade IN ('N', 'P', 'C', 'D', 'HD')),
    CONSTRAINT enrol_pk PRIMARY KEY (stu_nbr, unit_code,
enrol_year, enrol_semester)
);

-- You CANNOT use alter for the CHECKs, as the question said
'Include in the create table statement…' --
```

```
-- Part ii
Refer to Oracle documentation for the exact terminology:
https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_
9003.htm
```

```
-- Part iii
Oracle's formal definition: 'A trigger is like a stored procedure
that Oracle Database invokes automatically whenever a specified
event occurs.'
https://docs.oracle.com/database/121/LNPLS/triggers.htm#LNPLS2001

Example use for auditing (not limited to this example) --
Auditors need to record all deletions to a MANAGER table to keep a
record of project management staff who have left the company.
Triggers can insert into an LEFT_MANAGER table if a MANAGER row is
deleted.
```

(5d) INTERMEDIATE Database Theory and Implementation [2*5 = 10 marks]

Provide short answers to the following questions.
You are not expected to code anything in (5d).

i. A colleague of yours who is an expert Java (Object-Oriented) programmer but not familiar with databases wants to understand the design of an entity in your database model. What is the best diagram type to use, as learnt in this unit, and why?

ii. Provide an example of a SOFT CRASH in a database and distinguish it from a HARD CRASH.

iii. What is the difference between the HAVING and WHERE clauses in SQL?

iv. What is DATABASE MIDDLEWARE? Provide an example.

v. Provide two properties of NOSQL which make it useful for large companies like Google.

*** Q5 ENDS HERE ***
*** End of sample exam ***

(5d) SAMPLE ANSWERS:
i. UML Class Diagram as learnt in the unit which accommodates database design notations. It is used by, and familiar to, OO programmers.

ii. Refer Lecture 8. "Soft crashes - loss of volatile storage, but no damage to disks. These necessitate restart facilities."
Example: Blue Screen, etc.
"Hard crashes - anything that makes the disk permanently unreadable. These necessitate recovery facilities."
Example: dropped hard disk.

iii. Refer Lecture 9. "The WHERE clause is applied to ALL rows in the table. The HAVING clause is applied to the groups defined by the GROUP BY clause."

iv. Refer Lecture 11. "The DATABASE MIDDLEWARE — manages connectivity and data transformation issues." Example: MS ODBC.

v. Refer Lecture 12. Property 1: distributed (cluster friendly) - large clusters used by large companies. Property 2: No update,

append only, optimised for a 'main' operation e.g. lots of searches by Google users.