

Questions attempted: 0/11



A company wishes to record the following attributes about their employees: employee 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 Phyton

Use this data to explain the difference between a simple attribute, a composite attribute and a multivalued attribute. Your answer must include examples drawn from this data.

1 of 11



Report question issue Notes

### Question 1 Notes

☐ Unsure

Marks

The following relations represent a publications database:

AUTHOR(author\_id, author\_firstname, author\_lastname)

AUTHOR\_PAPER(author\_id, paper\_id, authorpaper\_position)

PAPER(paper\_id, paper\_title, journal\_id)

JOURNAL(journal\_id, journal\_title, journal\_month, journal\_year, journal\_editor)

\* journal editor in JOURNAL references AUTHOR(author\_id) – this is an author acting as the journal editor

Authors write papers which are published in an edition of a journal. Each edition of a journal is assigned a journal id and appoints an editor. A given paper may be authored by several authors, in such cases each author is assigned a position representing their contribution to the paper.

Write the relational algebra for the following queries (your answer must show an understanding of query efficiency):

List of symbols for copying/pasting as you enter your answers below:

project:  $\pi$ , select:  $\sigma$ , join:  $\bowtie$ , left outer join  $\ltimes$ , right outer join  $\rtimes$ , full outer join  $\ltimes\rtimes$ , intersect  $\cap$ , union  $\cup$ , minus

(a) Show the paper title, journal title, and month and year of journal publication for all papers published before 2012. (3 marks)

(b) Show the names of all authors who have never been listed as first author (authorpaper\_position = 1) in any paper. (4 marks)



List of symbols for copying/pasting as you enter your answers below:

project:  $\pi$ , select:  $\sigma$ , join:  $\bowtie$ , left outer join  $\ltimes$ , right outer join  $\rtimes$ , full outer join  $\ltimes\rtimes$ , intersect  $\cap$ , union  $\cup$ , minus

(a)

(b)

Report question issue

Notes

Question 2 Notes

2 of 11

☐ Unsure

Notification

Monash Computing Students Society (MCSS) is one of the student clubs at Monash University.

Students are welcome to join as a member. When a student joins MCSS, a member id is assigned, and the student's first name, last name, date of birth, email and phone number will be recorded. This club has an annual membership fee. When a member has paid the membership fee for the current year, the current year is recorded against the year of membership as part of their membership details.

MCSS hosts several events throughout the year. The events are currently categorised into *Professional Events*, *General Events*, and *Social Events*. MCSS would like to be able to add further categories as they develop new events. When an event is scheduled, MCSS assigns an event id to the event. The event date and time, description, location, allocated budget, the ticket price and the discount rate (e.g. 5%) for members. Some events are organised as free events for members. In this situation, the discount rate is recorded as 100% for members. For all events, only members can purchase the tickets. However, members can buy additional tickets for their friends or family at full price. For each of the sales, the receipt number, number of tickets sold, total amount paid and the member id are recorded.

Some events attract some sponsorships. The sponsor may be an organisation or an individual. The sponsors provide financial support to the event. Some events may have several sponsors. The amount of financial support provided by each sponsor is recorded for the event. Each sponsor is identified by a sponsor id. The name, contact email and sponsor type are also recorded. A sponsor may support several events throughout the year.

For some events such as career night, MCSS may also invite some guest speakers to share their experience. The database records all guests' information, the guests full name, email and phone number are recorded. If a guest comes from an organisation or an individual that provides a sponsorship to any of the MCSS events (does not have to be at the event where the guest speaks), this fact will also be recorded. A guest may be invited to several events.

Create a **logical level diagram** using Crow's foot notations to represent the "Monash Computing Students Society" data requirements described above. Clearly state any assumptions you make when creating the model.

Please note the following points:

- Be sure to include all relations, attributes and relationships (unnecessary relationships must not be included)
- Identify clearly the Primary Keys (P) and Foreign Keys (F), as part of your design
- In building your model you must conform to FIT2094 modelling requirements
- The following are **NOT** required on your diagram
  - verbs/names on relationship lines
  - indicators (\*) to show if an attribute is required or not
  - data types for the attributes

**NOTE:** This question has been designed such that the model will fit on a single A4 page. You are allowed to use two blank worksheets to draft your model and then submit your final response on ONE page.

The Super Electronics Invoice shown below displays the details of an invoice for the client Alice Paul.

ItemID	Item Name	Purchase Price	Expected Delivery Date	Quantity	Cost
316772	Soniq S55UV16B 55"	499.00	2 weeks	1	499.00
452550	Microsoft Surface Pro	1198.00	1-3 weeks	1	1198.00
483041	Delonghi Digital Coffee	299.00	Same Day	2	598.00
<b>SUB TOTAL:</b>					<b>\$ 2295.00</b>
<b>DELIVERY:</b>					<b>\$145.00</b>
<b>ORDER TOTAL:</b>					<b>\$2440.00</b>

Represent this form in UNF. In creating your representation you should note that Super Electronics wish to treat the client name and address as simple attributes. Convert your UNF to first normal form (1NF) and then continue the normalisation to third normal form (3NF). At each normal form show the appropriate dependencies for that normal form, if there are none write "No Dependencies".

Do not add new attributes during the normalisation. Clearly write the relations in each step from the unnormalised form (UNF) to the third normal form (3NF). Clearly, indicate primary keys on all relations from 1NF onwards.



Report question issue

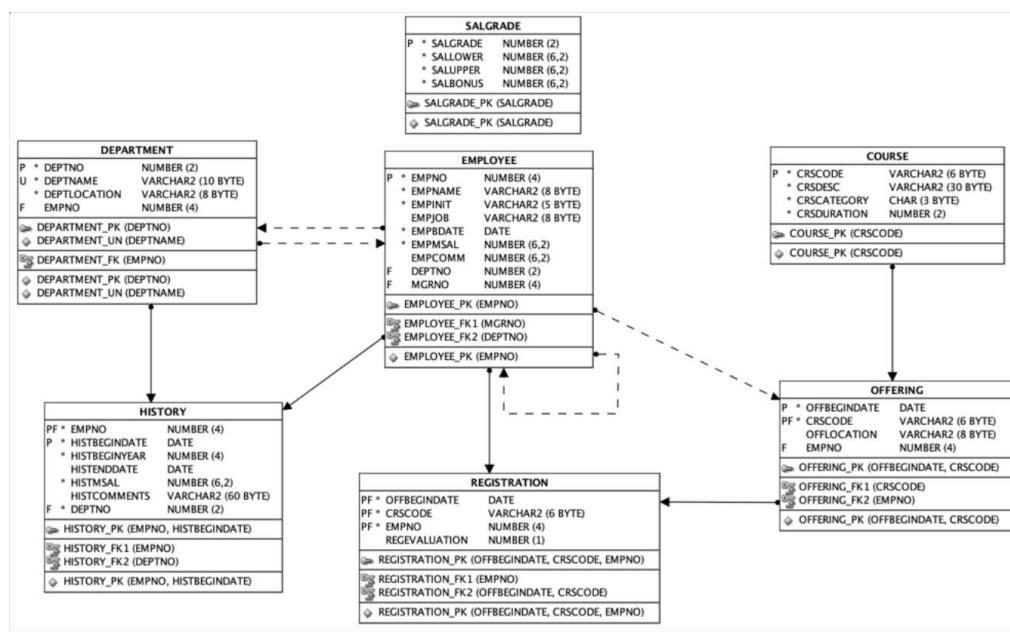
Notes

Question 4 Notes

Please open the below PDF with information on the case study to reference in a new tab:

### Employee System Model and Schema File for Part D

The following relational model depicts an employee system:



The schema file to create these tables is:

```
CREATE TABLE SALGRADE (
    salgrade NUMBER(2) NOT NULL ,
    sallower NUMBER(6,2) NOT NULL ,
    salupper NUMBER(6,2) NOT NULL ,
    salbonus NUMBER(6,2) NOT NULL ,
    CONSTRAINT salgrade_pk PRIMARY KEY (salgrade),
    CONSTRAINT salgrade_chk1 CHECK (sallower >= 0),
    CONSTRAINT salgrade_chk2 CHECK (sallower <= salupper));
```

```
COMMENT ON COLUMN salgrade.salgrade IS 'Salary Grade';
COMMENT ON COLUMN salgrade.sallower IS 'Salary Lower Limit';
COMMENT ON COLUMN salgrade.salupper IS 'Salary Upper Limit';
COMMENT ON COLUMN salgrade.salbonus IS 'Salary Bonus';
```

```
CREATE TABLE course (
    crscode VARCHAR(6) NOT NULL ,
    crsdesc VARCHAR(30) NOT NULL ,
    crscategory CHAR(3) NOT NULL ,
    crsduration NUMBER(2) NOT NULL ,
    CONSTRAINT course_pk PRIMARY KEY (crscode),
    CONSTRAINT course_chk1 CHECK (crscode = upper(crscode)),
    CONSTRAINT course_chk2 CHECK (crscategory in ('GEN','BLD','DSG')));
```

```
COMMENT ON COLUMN course.crscode IS 'Course Code';
COMMENT ON COLUMN course.crsdesc IS 'Course Description';
COMMENT ON COLUMN course.crscategory IS 'Course Category';
COMMENT ON COLUMN course.crsduration IS 'Course Duration';
```

```
deptno NUMBER(2) NOT NULL ,
deptname VARCHAR(10) NOT NULL ,
deptlocation VARCHAR(8) NOT NULL ,
empno NUMBER(4) ,
CONSTRAINT department_pk PRIMARY KEY (deptno),
CONSTRAINT department_un UNIQUE (deptname),
CONSTRAINT department_chk1 CHECK (deptname = upper(deptname)),
CONSTRAINT department_chk2 CHECK (deptlocation = upper(deptlocation));
```

```
COMMENT ON COLUMN department.deptno IS 'Department Number';
COMMENT ON COLUMN department.deptname IS 'Department Name';
COMMENT ON COLUMN department.deptlocation IS 'Location of department';
COMMENT ON COLUMN department.empno IS 'Employee who manages department';
```

```
CREATE TABLE EMPLOYEE (
empno NUMBER(4) NOT NULL ,
empname VARCHAR(8) NOT NULL ,
empinit VARCHAR(5) NOT NULL ,
empjob VARCHAR(8) ,
empbdate DATE NOT NULL ,
empmsal NUMBER(6,2) NOT NULL ,
empcomm NUMBER(6,2) ,
deptno NUMBER(2) ,
mgrno NUMBER(4) ,
CONSTRAINT employee_pk PRIMARY KEY (empno),
CONSTRAINT employee_fk1 FOREIGN KEY (mgrno)
REFERENCES EMPLOYEE (empno),
CONSTRAINT employee_fk2 FOREIGN KEY (deptno)
REFERENCES DEPARTMENT (deptno));

COMMENT ON COLUMN employee.empno IS 'Employee number';
COMMENT ON COLUMN employee.empname IS 'Employee name';
COMMENT ON COLUMN employee.empinit IS 'Employee initials';
COMMENT ON COLUMN employee.empjob IS 'Employee job';
COMMENT ON COLUMN employee.empbdate IS 'Employee birthdate';
COMMENT ON COLUMN employee.empmsal IS 'Employee monthly salary';
COMMENT ON COLUMN employee.empcomm IS 'Employee commission';
COMMENT ON COLUMN employee.deptno IS 'Department Number';
COMMENT ON COLUMN employee.mgrno IS 'Employees manager (empno of manager)';
```

```
ALTER TABLE DEPARTMENT
ADD (CONSTRAINT department_fk FOREIGN KEY (empno)
REFERENCES employee (empno));
```

```
empno NUMBER(4) NOT NULL ,
histbegindate DATE NOT NULL ,
histbeginyear NUMBER(4) NOT NULL ,
histenddate DATE ,
histmsal NUMBER(6,2) NOT NULL ,
histcomments VARCHAR(60) ,
deptno NUMBER(2) NOT NULL ,
CONSTRAINT history_pk PRIMARY KEY (empno, histbegindate),
CONSTRAINT history_chk CHECK (histbegindate < histenddate),
CONSTRAINT history_fk1 FOREIGN KEY (empno)
REFERENCES EMPLOYEE (empno)
ON DELETE CASCADE,
CONSTRAINT history_fk2 FOREIGN KEY (deptno)
REFERENCES DEPARTMENT (deptno));
```

```
COMMENT ON COLUMN history.deptno IS 'Department Number';
COMMENT ON COLUMN history.histbegindate IS 'Date history record begins';
COMMENT ON COLUMN history.histbeginyear IS 'Year history record begins';
COMMENT ON COLUMN history.histenddate IS 'Date history record ends';
COMMENT ON COLUMN history.histmsal IS 'Monthly Salary for this history
record';
COMMENT ON COLUMN history.histcomments IS 'Comments for this history record';
COMMENT ON COLUMN history.empno IS 'Employee number';
```

```
CREATE TABLE OFFERING (
offbegindate DATE NOT NULL ,
crscode VARCHAR(6) NOT NULL ,
offlocation VARCHAR(8) ,
empno NUMBER(4) ,
CONSTRAINT offering_pk PRIMARY KEY (offbegindate, crscode),
CONSTRAINT offering_fk1 FOREIGN KEY (crscode)
REFERENCES course(crscode),
CONSTRAINT offering_fk2 FOREIGN KEY (empno)
REFERENCES EMPLOYEE (empno));
```

```
COMMENT ON COLUMN offering.offbegindate IS 'Begin date for offering';
COMMENT ON COLUMN offering.crscode IS 'Course Code';
COMMENT ON COLUMN offering.offlocation IS 'Location for offering';
COMMENT ON COLUMN offering.empno IS 'Employee number for employee running
offering';
```

```
CREATE TABLE REGISTRATION (
offbegindate DATE NOT NULL ,
crscode VARCHAR(6) NOT NULL ,
empno NUMBER(4) NOT NULL,
revaluation NUMBER(1) ,
CONSTRAINT registration_pk PRIMARY KEY (offbegindate, crscode, empno),
CONSTRAINT resgitration_chk CHECK (revaluation in (1,2,3,4,5)),
CONSTRAINT registration_fk1 FOREIGN KEY (empno)
REFERENCES EMPLOYEE (empno),
CONSTRAINT registration_fk2 FOREIGN KEY (offbegindate, crscode)
REFERENCES OFFERING (offbegindate, crscode));
```



```
COMMENT ON COLUMN registration.crscode      IS 'Course Code';
COMMENT ON COLUMN registration.reevaluation IS 'Grade for course completed';
COMMENT ON COLUMN registration.empno        IS 'Employee number of employee
completing course';
```

☐ Unsure

Questions attempted: 0/11

10  
Marks

The company needs to record a new department. This new department's number will be 10 higher than the highest current department number and will be called EXAM and is located in BOSTON. The employee named KING who has a job as the only company DIRECTOR has been assigned to manage the new EXAM department.

The company has also decided that they wish to record, for each department, the number of employees currently working in the department (the employee count). For new departments the number of employees in the department should be set to 0. For those departments which currently have employees, the employee count should correctly reflect the current number of employees in the department.

Code the SQL statements to modify the database to meet these requirements.

**5** of 11Report question issue Notes 

Question 5 Notes

☐ Unsure



Questions attempted: 0/11

6  
Marks

---	---	---	---	---	--

List the employee number, the employee name, the employee job and the yearly salary of all employees that belong to the 'Sales' department. The name of the employee must be shown in a column called "Employee Name" and the yearly salary must be shown in the form of \$34,200 in a column called "Yearly Salary". Show the employee with the highest salary first, if two employees have the same salary, order it based on the employee number.

Code the SQL SELECT statement.

A ▼
B
*I*
U
 $x_2$ 
 $x^2$ 




 $\frac{1}{2}$   
 $\frac{3}{5}$ 







6 of 11

Report question issue  Notes 

### Question 6 Notes

☐ Unsure



Questions attempted: 0/11

9  
Marks

For each course which has been completed by at least 5 employees, list the course code, the course description and the course duration. The course duration must be shown in a column called "Course Duration" and include the word 'days' (e.g. 4 days). Order the output by the course code.

Code the SQL SELECT statement.

A ▼ B I U x<sub>2</sub> x<sup>2</sup> [List Icons] [Table Icon] [ABC 123] [Undo] [Redo] [Fullscreen]

7 of 11

Report question issue 

Notes ○

### Question 7 Notes

☐ Unsure



Questions attempted: 0/11

15  
Marks

---	---	---	---	---	--

List ALL employees whose total course registrations are less than the average number of registrations for employees who have registered for a course. Note that some employees may repeat a course, this repeat does not count as a different course. In the list, include the employee number, name, date of birth and the number of different courses they have registered for. Order the output by employee number.

Code the SQL SELECT statement.

THE UNIVERSITY OF CHICAGO PRESS

A ▼ B I U x<sub>2</sub> x<sup>2</sup>    ≡ ≡ ≡    ≡ ≡ ≡    1/2 2/3    ≡ ≡       ABC 123    ↺ ↻    ↶ ↷ ↸ ↹

8 of 11

Report question issue 

Notes ○

### Question 8 Notes

☐ Unsure

We wish to develop a php based web page which shows all departments, and their manager's name and monthly salary as shown below:

### Department List - PAYROLL Database

Department Number	Department Name	Department Location	Manager	Manager's Monthly Salary
10	ACCOUNTING	NEW YORK	AB CLARK	\$2450.00
20	TRAINING	DALLAS	JM JONES	\$2975.00
30	SALES	CHICAGO	R BLAKE	\$2850.00
40	HR	BOSTON	CC KING	\$5000.00

Here is the incomplete PHP code for the page:

```
<table border =1 width=1000>
    <tr>
        <th width=200><b>Department Number</b></th>
        <th width=200><b>Department Name</b></th>
        <th width=200><b>Department Location</b></th>
        <th width=200><b>Manager</b></th>
        <th width=200><b>Manager's Monthly Salary</b></th>
    </tr>
```

```
<?php
    $query = complete this part!;
```

```
$stmt = oci_parse($conn,$query);
if (!$stmt) {
    $e = oci_error($conn);
    print "Error on parse of statement:<br>" ;
    print $e['message'] ;
    exit;
}

oci_define_by_name($stmt,"DNO",$dno);
oci_define_by_name($stmt,"DNAME",$dname);
oci_define_by_name($stmt,"DLOC",$dloc);
oci_define_by_name($stmt,"MGRNAME",$mgrname);
oci_define_by_name($stmt,"MGRMSAL",$mgrmsal);

$r = oci_execute($stmt);
if (!$r) {
    $e = oci_error($stmt);
    print "Error execute of statement:<br>" ;
    print $e['message'] ;
    exit;
}

while (oci_fetch($stmt)) {
    print("
    <tr>
        <td width=200>$dno</td>
        <td width=200>$dname</td>
        <td width=200>$dloc</td>
        <td width=200>$mgrname</td>
        <td width=200>$mgrmsal</td>
    </tr>");
}

print("</table>");
```



Questions attempted: 0/11



5

Marks

Given two transactions:

T1 – R(X), W(X)

T2 – R(Y), W(Y), R(X), W (X)

Where R(X) means Read(X) and W(X) means Write(X).

(a) If we wish to complete both of these transactions, explain the difference between a *serial* and *non-serial* ordering of these two transactions. Provide an example of each as part of your answer.

(b) What transaction ACID property does a non-serial ordering of these two transactions potentially violate.



(a)

(b)

Report question issue Notes 

Question 10 Notes

☐ Unsure

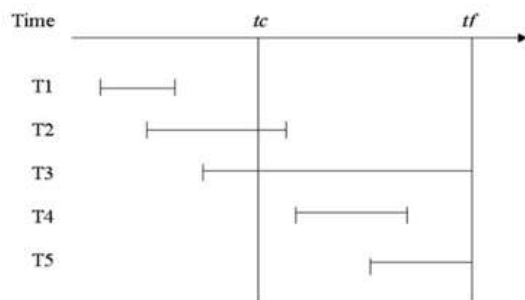


Questions attempted: 0/11



5  
Marks

A *write through* database has five transactions running as listed below (the time is shown horizontally from left to right):




At time *tc* a checkpoint is taken, at time *tf* the database fails due to a power outage.

Explain for each transaction what recovery operations will be needed when the database is restarted and why.

11 of 11



Report question issue  Notes 

Question 11 Notes

☐ Unsure