Database Assignment 3

Saeth Wannasuphoprasit
Student ID: 201585689

Question One (60 marks)

Consider the following relational database schema, where the underline attributes are the primary keys:

- Employee(eid, ename, age)
- Department(did, dname, dtype, address)
- WorksIn(eid, did, since)
- Product(pid, pname, ptype, pcolor)
- Sells(did, pid, quantity);

1. **(14 marks)** Create the above schemas in **MySQL**, using the **CREATE TABLE** statement. Make sure that you define all possible **keys**, and that **entity integrity** and **referential integrity** are guaranteed. Explain in detail any assumptions you may make.

Step 1: define primary keys, foreign keys, data type, entity integrity, and referential integrity. Also, define possible data to insert into the table.

employee(eid, ename, age)
primary key: eid (INT), not NULL, unique, no default value/ [1,2,3,...]
foreign key: no
ename (VARCHAR(45)), not NULL, not unique, no default value
/ [saeth, prang, champ]
age (INT), not NULL, not unique, no default value
/ [20, 21, 22, 30, 31, 32, 40, 41, 42]
Assumption:

1. There can be duplicated ename and age.

department(did, dname, dtype, address)
primary key: did (INT), not NULL, unique, no default value/ [1,2,3,...]
foreign key: no
dname (VARCHAR(45)), not NULL, not unique, no default value
/ [robinson, tesco, tkmax, Central]
dtype (VARCHAR(45)), not NULL, not unique, no default value
/ [tech, toys, food]
address (VARCHAR(45)), NULL, not unique
/[liverpool, everton, NULL]
Assumptions:

1. There can be duplicated dname, dtype, and address.
2. address can accept NULL values.

works_in(eid, did, since)
primary key 1: eid (INT), not NULL, not unique, no default value/ [1,2,3,...]
primary key 2: did (INT), not NULL, not unique, no default value/ [1,2,3,...]
foreign key 1: eid references employee(eid), ON DELETE CASCADE, ON UPDATE CASCAD
foreign key 2: did references department(did), ON DELETE CASCADE, ON UPDATE CASCAD
since (DATE), NULL, not unique, no default value
/[1990-09-01, 1990-09-02, 1990-09-03, NULL]
Assumption:

1. There can be duplicated eid, did, and since. But the combination of eid and did is unique.
2. We use CASCADE on both ON DELETE and ON UPDATE because we want to update any changes as soon as possible.
3. since accepts NULL values.

product(pid, pname, ptype, pcolor)
primary key: pid (INT), not NULL, unique, no default value/ [1,2,3,...]
foreign key: no
pname (VARCHAR(45)), not NULL, not unique, no default value
/ [tool1, tool2, toy1, toy2, wear1, wear2, gadget1, gadget2, drink1, drink2, snack1, snack2]
ptype (VARCHAR(45)), not NULL, not unique, no default value
/ [tool, toy, wear, gadget, drink, snack]
pcolor (VARCHAR(45)), NULL, not unique, no default value
/[blue, red, NULL]
Assumptions:

1. There can be duplicated pname, ptype, and pcolor.
2. pcolor can accept NULL values.

sells(did, pid, quantity)
primary key 1: did (INT), not NULL, not unique, no default value/ [1,2,3,...]
primary key 2: pid (INT), not NULL, not unique, no default value/ [1,2,3,...]
foreign key 1: did references department(eid), ON DELETE CASCADE, ON UPDATE CASCAD
foreign key 2: pid references product(did), ON DELETE CASCADE, ON UPDATE CASCAD
quantity (INT), not NULL, not unique, default value = 1
/[1, 2, 3]
Assumption:

1. There can be duplicated did, pid, and quantity. But the combination of did and pid is unique.
2. We use CASCADE on both ON DELETE and ON UPDATE because we want to update any changes as soon as possible.
3. The default value of quantity is 1.

Step2: according to the above criteria, we can write SQL to create tables and insert data in MySQL as follows.

1. **employee**

```
CREATE TABLE `database_assignment_3`.`employee` (
  `eid` INT NOT NULL,
  `ename` VARCHAR(45) NOT NULL,
  `age` INT NOT NULL,
  PRIMARY KEY (`eid`));

INSERT INTO `database_assignment_3`.`employee` (`eid`, `ename`, `age`) VALUES ('1', 'saeth', '20');
INSERT INTO `database_assignment_3`.`employee` (`eid`, `ename`, `age`) VALUES ('2', 'saeth', '21');
INSERT INTO `database_assignment_3`.`employee` (`eid`, `ename`, `age`) VALUES ('3', 'saeth', '22');
INSERT INTO `database_assignment_3`.`employee` (`eid`, `ename`, `age`) VALUES ('4', 'prang', '30');
INSERT INTO `database_assignment_3`.`employee` (`eid`, `ename`, `age`) VALUES ('5', 'prang', '31');
INSERT INTO `database_assignment_3`.`employee` (`eid`, `ename`, `age`) VALUES ('6', 'prang', '32');
INSERT INTO `database_assignment_3`.`employee` (`eid`, `ename`, `age`) VALUES ('7', 'champ', '40');
INSERT INTO `database_assignment_3`.`employee` (`eid`, `ename`, `age`) VALUES ('8', 'champ', '41');
INSERT INTO `database_assignment_3`.`employee` (`eid`, `ename`, `age`) VALUES ('9', 'champ', '42');
```

employee		
eid	ename	age
1	saeth	20
2	saeth	21
3	saeth	22
4	prang	30
5	prang	31
6	prang	32
7	champ	40
8	champ	41
9	champ	42

2. **department**

```
CREATE TABLE `database_assignment_3`.`department` (
  `did` INT NOT NULL,
  `dname` VARCHAR(45) NOT NULL,
  `dtype` VARCHAR(45) NOT NULL,
  `address` VARCHAR(45) NULL,
  PRIMARY KEY (`did`));
```

```
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('1', 'Central', 'tech', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('2', 'Central', 'toys', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('3', 'Central', 'food', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('4', 'Central', 'tech', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('5', 'Central', 'toys', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('6', 'Central', 'food', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES ('7',
'Central', 'tech');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES ('8',
'Central', 'toys');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES ('9',
'Central', 'food');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('10', 'tesco', 'tech', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('11', 'tesco', 'toys', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('12', 'tesco', 'food', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('13', 'tesco', 'tech', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('14', 'tesco', 'toys', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('15', 'tesco', 'food', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES
('16', 'tesco', 'tech');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES
('17', 'tesco', 'toys');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES
('18', 'tesco', 'food');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('19', 'tkmax', 'tech', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('20', 'tkmax', 'toys', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('21', 'tkmax', 'food', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('22', 'tkmax', 'tech', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('23', 'tkmax', 'toys', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`)
VALUES ('24', 'tkmax', 'food', 'everton');
```

```
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES ('25', 'tkmax', 'tech');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES ('26', 'tkmax', 'toys');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES ('27', 'tkmax', 'food');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`) VALUES ('28', 'robinson', 'tech', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`) VALUES ('29', 'robinson', 'toys', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`) VALUES ('30', 'robinson', 'food', 'liverpool');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`) VALUES ('31', 'robinson', 'tech', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`) VALUES ('32', 'robinson', 'toys', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`, `address`) VALUES ('33', 'robinson', 'food', 'everton');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES ('34', 'robinson', 'tech');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES ('35', 'robinson', 'toys');
INSERT INTO `database_assignment_3`.`department` (`did`, `dname`, `dtype`) VALUES ('36', 'robinson', 'food');
```

department			
did	dname	dtype	address
1	Central	tech	liverpool
2	Central	toys	liverpool
3	Central	food	liverpool
4	Central	tech	everton
5	Central	toys	everton
6	Central	food	everton
7	Central	tech	NULL
8	Central	toys	NULL
9	Central	food	NULL
10	tesco	tech	liverpool
11	tesco	toys	liverpool
12	tesco	food	liverpool
13	tesco	tech	everton
14	tesco	toys	everton
15	tesco	food	everton
16	tesco	tech	NULL
17	tesco	toys	NULL
18	tesco	food	NULL
19	tkmax	tech	liverpool
20	tkmax	toys	liverpool
21	tkmax	food	liverpool
22	tkmax	tech	everton
23	tkmax	toys	everton
24	tkmax	food	everton
25	tkmax	tech	NULL
26	tkmax	toys	NULL
27	tkmax	food	NULL
28	robinson	tech	liverpool
29	robinson	toys	liverpool
30	robinson	food	liverpool
31	robinson	tech	everton
32	robinson	toys	everton
33	robinson	food	everton
34	robinson	tech	NULL
35	robinson	toys	NULL
36	robinson	food	NULL

3. **works_in**

```
CREATE TABLE `database_assignment_3`.`works_in` (
  `eid` INT NOT NULL,
  `did` INT NOT NULL,
  `since` DATE NULL,
  PRIMARY KEY (`eid`, `did`),
  FOREIGN KEY (eid) REFERENCES employee(eid)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  FOREIGN KEY (did) REFERENCES department(did)
  ON DELETE CASCADE
  ON UPDATE CASCADE) ;
```

```
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('1', '10', '1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('1', '13', '1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('1', '16', '1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('1', '17', '1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('1', '20', '1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`) VALUES ('1', '25');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`) VALUES ('1', '35');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('2', '10', '1990-09-03');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('2', '13', '1990-09-03');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('2', '16', '1990-09-03');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('2', '17', '1990-09-03');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`) VALUES ('2', '20');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`) VALUES ('2', '25');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('2', '35', '1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('7', '1', '1990-09-03');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('7', '2', '1990-09-03');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('7', '5', '1990-09-03');
```

```
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('7', '7',
'1990-09-03');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('7', '10',
'1990-09-01');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('7', '13',
'1990-09-01');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('7', '16',
'1990-09-01');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('7', '17',
'1990-09-01');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`) VALUES ('7', '20');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('7', '25',
'1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`) VALUES ('7', '35');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('9', '1',
'1990-09-01');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('9', '2',
'1990-09-01');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('9', '5',
'1990-09-01');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('9', '7',
'1990-09-01');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('9', '10',
'1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('9', '13',
'1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('9', '16',
'1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('9', '17',
'1990-09-02');
INSERT INTO `database_assignment_3`.`works_in` (`eid`, `did`, `since`) VALUES ('9', '20',
'1990-09-02');
```

works_in		
eid	did	since
1	10	9/2/1990
1	13	9/2/1990
1	16	9/2/1990
1	17	9/2/1990
1	20	9/2/1990
1	25	NULL
1	35	NULL
2	10	9/3/1990
2	13	9/3/1990
2	16	9/3/1990
2	17	9/3/1990
2	20	NULL
2	25	NULL
2	35	9/2/1990
7	1	9/3/1990
7	2	9/3/1990
7	5	9/3/1990
7	7	9/3/1990
7	10	9/1/1990
7	13	9/1/1990
7	16	9/1/1990
7	17	9/1/1990
7	20	NULL
7	25	9/2/1990
7	35	NULL
9	1	9/1/1990
9	2	9/1/1990
9	5	9/1/1990
9	7	9/1/1990
9	10	9/2/1990
9	13	9/2/1990
9	16	9/2/1990
9	17	9/2/1990
9	20	9/2/1990

4. **product**

```
CREATE TABLE `database_assignment_3`.`product` (
  `pid` INT NOT NULL,
  `pname` VARCHAR(45) NOT NULL,
  `ptype` VARCHAR(45) NOT NULL,
  `pcolor` VARCHAR(45) NULL,
  PRIMARY KEY (`pid`));
```

```
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`, `pcolor`) VALUES ('1', 'tool1', 'tool', 'red');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`, `pcolor`) VALUES ('2', 'tool2', 'tool', 'blue');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`, `pcolor`) VALUES ('3', 'toy1', 'toy', 'blue');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`) VALUES ('4', 'toy2', 'toy');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`, `pcolor`) VALUES ('5', 'wear1', 'wear', 'red');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`, `pcolor`) VALUES ('6', 'wear2', 'wear', 'blue');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`, `pcolor`) VALUES ('7', 'gadget1', 'gadget', 'blue');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`) VALUES ('8', 'gadget2', 'gadget');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`, `pcolor`) VALUES ('9', 'drink1', 'drink', 'red');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`, `pcolor`) VALUES ('10', 'drink2', 'drink', 'red');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`, `pcolor`) VALUES ('11', 'snack1', 'snack', 'blue');
INSERT INTO `database_assignment_3`.`product` (`pid`, `pname`, `ptype`, `pcolor`) VALUES ('12', 'snack2', 'snack', 'red');
```

product			
pid	pname	ptype	pcolor
1	tool1	tool	red
2	tool2	tool	blue
3	toy1	toy	blue
4	toy2	toy	NULL
5	wear1	wear	red
6	wear2	wear	blue
7	gadget1	gadget	blue
8	gadget2	gadget	NULL
9	drink1	drink	red
10	drink2	drink	red
11	snack1	snack	blue
12	snack2	snack	red

5. **sells**

```
CREATE TABLE `database_assignment_3`.`sells` (
  `did` INT NOT NULL,
  `pid` INT NOT NULL,
  `quantity` INT NOT NULL DEFAULT '1',
  PRIMARY KEY (`did`, `pid`),
  INDEX `pid_idx` (`pid` ASC) VISIBLE,
  CONSTRAINT `did`
    FOREIGN KEY (`did`)
    REFERENCES `database_assignment_3`.`department` (`did`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `pid`
    FOREIGN KEY (`pid`)
    REFERENCES `database_assignment_3`.`product` (`pid`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);
```

```
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`, `quantity`) VALUES ('25', '2', '3');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`, `quantity`) VALUES ('25', '7', '3');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('25', '5');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('35', '7');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`, `quantity`) VALUES ('9', '3', '2');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('9', '5');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('9', '7');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`, `quantity`) VALUES ('9', '9', '2');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('9', '11');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`, `quantity`) VALUES ('10', '1', '3');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('10', '4');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`, `quantity`) VALUES ('10', '5', '2');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('10', '8');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('10', '9');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('20', '1');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`, `quantity`) VALUES ('20', '3', '3');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('20', '5');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('20', '7');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('20', '9');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`, `quantity`) VALUES ('20', '12',
'3');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('34', '1');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('34', '2');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('34', '3');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`, `quantity`) VALUES ('34', '8', '2');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('1', '5');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('1', '6');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`, `quantity`) VALUES ('1', '9', '2');
INSERT INTO `database_assignment_3`.`sells` (`did`, `pid`) VALUES ('1', '10');
```

sells		
did	pid	quantity
1	5	1
1	6	1
1	9	2
1	10	1
9	3	2
9	5	1
9	7	1
9	9	2
9	11	1
10	1	3
10	4	1
10	5	2
10	8	1
10	9	1
20	1	1
20	3	3
20	5	1
20	7	1
20	9	1
20	12	3
25	2	3
25	5	1
25	7	3
34	1	1
34	2	1
34	3	1
34	8	2
35	7	1

2. Provide **MySQL** queries for the following:

(a) **(3 marks)** Find the names of departments which sell blue products.

```
SELECT DISTINCT dname
FROM database_assignment_3.department NATURAL JOIN database_assignment_3.sells NATURAL JOIN database_assignment_3.product
WHERE product.pcolor = 'blue';
```

Result Grid

	dname
▶	tkmax
	robinson
	Central

In this case, we select dname from the natural join of department, sells, and product with the condition pcolor = 'blue'.

(b) **(6 marks)** Find the names of departments which sell products of type tool and products of type toy.

```
(SELECT DISTINCT dname
FROM database_assignment_3.department NATURAL JOIN database_assignment_3.sells
NATURAL JOIN database_assignment_3.product
WHERE product.ptype = 'tool')
```

INTERSECT

```
(SELECT DISTINCT dname
FROM database_assignment_3.department NATURAL JOIN database_assignment_3.sells
NATURAL JOIN database_assignment_3.product
WHERE product.ptype = 'toy')
```

Result Grid

	dname
▶	tesco
	tkmax
	robinson

1. select dname from the natural join of department, sells, and product with the condition ptype = 'tool' -> {'tesco', 'tkmax', 'robinson'}
2. select dname from the natural join of department, sells, and product with the condition ptype = 'toy' -> {'tesco', 'tkmax', 'robinson', 'Central'}
3. INTERSECT 1 and 2, we will get the result which is {'tesco', 'tkmax', 'robinson'}

(c) **(8 marks)** Find the names of departments which sell blue products and do not have any employee older than 40.

```
SELECT DISTINCT dname
FROM database_assignment_3.department NATURAL JOIN database_assignment_3.sells
NATURAL JOIN database_assignment_3.product
WHERE product.pcolor = 'blue' AND did IN (SELECT DISTINCT did from database_assignment_3.works_in) AND
did NOT IN (SELECT DISTINCT did
FROM database_assignment_3.employee NATURAL JOIN database_assignment_3.works_in
WHERE employee.age > 40)
```

1. select dname from the natural join of department, sells, and product with the condition pcolor = ‘blue’ -> {‘Central’, ‘tkmax’, ‘robinson’}
2. fillter did in works_in (to make sure that did has employee)
3. fillter out did in the natural join employee and works_in, where age > 40 to get rid of all did which has age > 40
4. finally, we will have the desired result

Result Grid

	dname
▶	tkmax
	robinson

(d) **(9 marks)** For each department report the department-id and the age of the oldest employee working in it.

```
SELECT did, max(age)
FROM database_assignment_3.employee NATURAL JOIN database_assignment_3.works_in
GROUP BY did;
```

Result Grid

	did	max(age)
▶	1	42
	2	42
	5	42
	7	42
	10	42
	13	42
	16	42
	17	42
	20	42
	25	40
	35	40

1. select did and max(age) from the natural join of employee and works_in
2. group the result by did

Note that not all did in department have employee. So, in this case, the result will show only department id that has at least 1 employee and display the maximum age of all employee in the target department. The id of the rest department means that those departments don't have any workers in it. So, there is no maximum age and it is not appripriate to display in the result.

(e) **(9 marks)** Find the names of employees who are older than at least one employee working in department 'Central'.

```
SELECT DISTINCT ename
FROM database_assignment_3.employee
WHERE age > (SELECT min(age) FROM database_assignment_3.department NATURAL JOIN database_assignment_3.works_in
NATURAL JOIN database_assignment_3.employee
WHERE dname = 'Central');
```

Result Grid

	ename
▶	champ

This question can be interpreted as follows.
select ename from employee, where age is more than the minimum of the employee working in Central department.
So, we can find the minimum of employee working in Cantral by using min(age) from the natural join of department and works_in, where dname is Central. Then, we select dname only if their age is more than the min(age) computed before.

(f) **(11 marks)** Find the names of employees working in departments which have sold at least 5 types of products.

```
SELECT DISTINCT ename
FROM database_assignment_3.employee NATURAL JOIN database_assignment_3.works_in
WHERE did IN (SELECT did
FROM database_assignment_3.product NATURAL JOIN database_assignment_3.sells
GROUP BY did
HAVING count(DISTINCT ptype) >= 5)
```

Result Grid

	ename
▶	saeth
	champ

1. find did in the natural join of product and sells, group by did with the counting of ptype >= 5 to get the did that has at least 5 product types
2. link those did to the ename by natural join of the table employee and works_in to find the ename

Question Two (30 marks)

Assume that there are three transactions T_1, T_2, T_3 that operate (read and write) on the data items A, B, and C. We are using the following notation: RJ(X) means that the transaction T_J reads the data item X, while WJ(X) means that the transaction T_J writes on the data item X. For example R1(A) means that the transaction T_1 reads the data item A, i.e., $read(T_1, A)$, while W3(B) would mean that the transaction T_3 writes on the data item B, i.e., $write(T_3, B)$.

You are given the following schedules S1, S2

1. S1: R1(A),R1(B),W1(A),R2(A),R1(C),W1(C),R3(C),W2(A),R3(B),W3(A)
2. S2: R1(A),W1(A),R2(C),R2(B),R3(C),W3(C),R3(A),R1(B),W1(B),R1(C)
3. S3: R1(A),R1(B),W1(A),R2(A),W3(C),W1(C),W2(A)

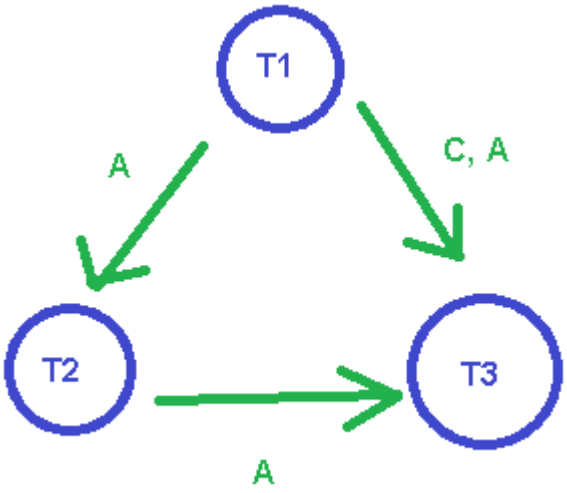
   For each of the above schedules

(i) **(3 marks)** create the precedence graph of the conflicts.

we can write S1 as follows.

S1		
T1	T2	T3
read(A)		
read(B)		
write(A)		
	read(A)	
read(C)		
write(C)		
		read(C)
	write(A)	
		read(B)
		write(A)

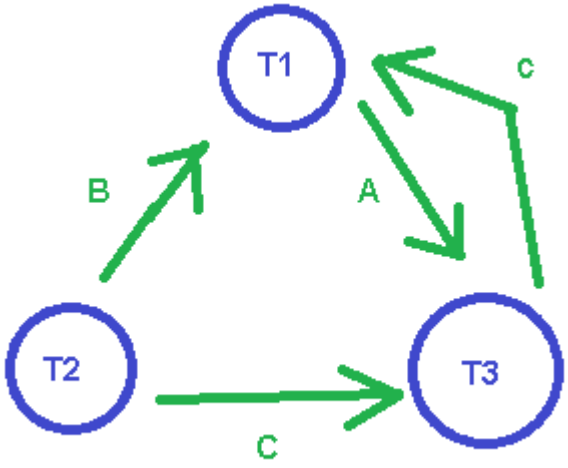
T2:read(A) after T1:write(A)
T3: write(A) after T2:write(A)
T3:read(C) after T1:write(C)
T3:write(A) after T1:write(A)
So, the precedence graph is as follows.



we can write S2 as follows.

S2		
T1	T2	T3
read(A)		
write(A)		
	read(C)	
	read(B)	
		read(C)
		write(C)
		read(A)
read(B)		
write(B)		
read(C)		

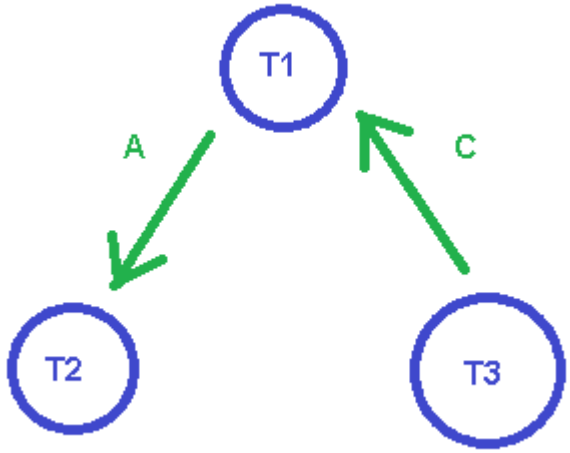
T2:write(B) after T2:read(B)
T3:write(C) after T2:read(C)
T1:read(C) after T3:write(C)
T3:read(A) after T1:write(A)
So, the precedence graph is as follows.



we can write S3 as follows.

S3		
T1	T2	T3
read(A)		
read(B)		
write(A)		
	read(A)	
		write(C)
write(C)		
	write(A)	

T2:read(A) after T1:write(A)
T1:write(C) after T3:write(C)
So, the precedence graph is as follows.



(ii) **(2 marks)** show whether the schedule is conflict-serializable or not. In case it is conflict-serializable, show a corresponding serial schedule. In case it is **not** conflict-serializable, explain shortly why this is the case.

S1:
conflic-serialization (T1 -> T2 -> T3 in the order of A and A path in the S1 precedence graph)

S2:
non conflic-serialization because there is a cycle in T1 and T3 (A and C path in the S2 precedence graph)

S3:
conflic-serialization (T3 -> T1 -> T2 in the order of C and A path in the S3 precedence graph)
Note that this is also serial schedue T3 -> T1 -> T2

(iii) **(5 marks)** can this schedule occur by use of (two-phase locking) 2PL? Explain your answer.

S1: can occur with 2PL by the illustration below.

S1		
T1	T2	T3
write_lock(A)		
read(A)		
read_lock(B)		
read(B)		
write(A)	write_lock(A)	
write_lock(C)	wait	
unlock(A, B)	wait	
	read(A)	
read(C)		
write(C)		read_lock(C)
unlock(C)		wait
		read(C)
	write(A)	read_lock(B)
	unlock(A)	read(B)
		write_lock(A)
		write(A)
		unlock(C, B, A)

S2: can occur with 2PL by the illustration below.

S2		
T1	T2	T3
write_lock(A)		
read(A)		
write(A)	read_lock(C)	
	read(C)	
	read_lock(B)	
	read(B)	write_lock(C)
	unlock(C, B)	wait
write_lock(B)		read(C)
read_lock(C)		write(C)
unlock(A)		read_lock(A)
		read(A)
read(B)		unlock(C, A)
write(B)		
read(C)		
unlock(C, B)		

S3: can not occur with 2PL by the illustration below.

S3		
T1	T2	T3
write_lock(A)		
read(A)		
read_lock(B)		
read(B)		
write(A)	write_lock(A)	
write_lock(C)	wait	
unlock(A)	wait	
	read(A)	write_lock(C)
		wait
		wait
can not unlock C		wait

S3		
T1	T2	T3
read(A)		
read(B)		
write(A)		
	read(A)	
		write(C)
write(C)		
	write(A)	

write(C) in T3 can not be done because if we unlock C in T1, we can not give write_lock(C) in T1 after that.

Question Three (10 marks)

Consider the following transactions T_1 and T_2

Time	T_1	T_2
1	read item(A)	
2	A=A-2	
3		product = 1
4		read item(A)
5	write item(A)	
6		product = product*A
7		A=A-1
8	read item(B)	
9	B=B+1	
10	write item(B)	
11	commit	
12		write item(A)
13		read item(B)
14		B=B+1
15		product = product*B
16		commit

At time step 0 the value of A is 3 and B is 5.

1. **(8 marks)** What are the values of the data items A and B after time step 16? What value does the "product" have[1]? You should give a table, having the values of the data items in the database at each time step, as well as the value of the local variable "product". We assume that the local variable "product" **doesn't have a value** before the time step 3. Your solution should start like in the following table.

[1]Note that "product" is a local variable of the transaction, that does not necessarily exist in the database.

Time	A	B	product
0	3	5	n/a
1	3	5	n/a
⋮	⋮	⋮	⋮

Time	Action	Action result	A	B	product (local variable)
0	-	-	3	5	n/a
1	T1 read A	T1 read 3 from A	3	5	n/a
2	T1: A = A -2	T1: A = 3 - 2 = 1, and not write to A	3	5	n/a
3	T2: product = 1	product = 1 (local variable)	3	5	1
4	T2 read A	T 2 read 3 from A	3	5	1
5	T1 write A	T 1 write 1(Time = 2) in A	1	5	1
6	T2: product = product * A	product = 1 * 3 (Time = 4, T2 read 3) = 3	1	5	3
7	T2: A = A - 1	T2: A = 3(Time = 4, T2 read 3) - 1 = 2, and not write to A	1	5	3
8	T1 read B	T1 read 5 from B	1	5	3
9	T1: B = B + 1	T1: B = 5(Time = 8, T1 read 5) + 1 = 6, and not write to B	1	5	3
10	T1 write B	T1 write 6(from Time = 9) to B	1	6	3
11	T1 commit	T1 commit	1	6	3
12	T2 write A	T2 write 2(from Time = 7) to A	2	6	3
13	T2 read B	T2 read 6 from B	2	6	3
14	T2: B = B + 1	T2: B = 6(Time = 13, T2 read 6) + 1 = 7, and not write to B	2	6	3
15	T2: product = product * B	T2: product = 3 * 7(Time = 14) = 21	2	6	21
16	T2 commit	T2 commit	2	6	21

So, after time step 16, A = 2, B = 6, and product = 21

2. **(1 marks)** What are the **final** values of the data items A and B if we first execute T_1, and then T_2? What **final** value does the "product" have?

Time	Action	Action result	A	B	product (local variable)
0	-	-	3	5	n/a
1	T1 read A	T1 read 3 from A	3	5	n/a
2	T1: A = A -2	T1: A = 3 - 2 = 1, and not write to A	3	5	n/a
3	T1 write A	T1 write 1(Time = 2) to A	1	5	n/a
4	T1 read B	T1 read 5 from B	1	5	n/a
5	T1: B = B + 1	T1: B = 5(at Time = 4) + 1 = 6, and not write to B	1	5	n/a
6	T1 write B	T1 write 6(at Time = 5) to B	1	6	n/a
7	T1 commit	T1 commit	1	6	n/a
8	T2: product = 1	product = 1 (local variable)	1	6	1
9	T2 read A	T2 read 1 from A	1	6	1
10	T2: product = product * A	T2: product = 1 * 1(A at Time = 9) = 1	1	6	1
11	T2: A = A - 1	T2: A = 1(at Time = 9) - 1 = 0, and not write to A	1	6	1
12	T2 write A	T2 write 0(from Time = 11) to A	0	6	1
13	T2 read B	T2 read 6 from B	0	6	1
14	T2: B = B + 1	T2: B = 6(from Time = 13) + 1 = 7, and not write to B	0	6	1
15	T2: product = product * B	T2: product = 1 * 7(from Time = 14) = 7	0	6	7
16	T2 commit	T2 commit	0	6	7

So, after time step 16, A = 0, B = 6, and product = 7

3. **(1 marks)** What are the **final** values of the data items A and B if we first execute T_2, and then T_1? What **final** value does the "product" have?

Time	Action	Action result	A	B	product (local variable)
0	-	-	3	5	n/a
1	T2: product = 1	product = 1 (local variable)	3	5	1
2	T2 read A	T2 read 3 from A	3	5	1
3	T2: product = product * A	T2: product = 1 * 3(A at Time = 2) = 3	3	5	3
4	T2: A = A - 1	T2: A = 3(from Time = 2) - 1 = 2, and not write to A	3	5	3
5	T2 write A	T2 write 2(from Time = 4) to A	2	5	3
6	T2 read B	T2 read 5 from B	2	5	3
7	T2: B = B + 1	T2: B = 5(from Time = 6) + 1 = 6, and not write to B	2	5	3
8	T2: product = product * B	T2: product = 3 * 6(B from Time = 7) = 18	2	5	18
9	T2 commit	T2 commit	2	5	18
10	T1 read A	T1 read 2 from A	2	5	18
11	T1: A = A -2	T1: A = 2(from Time = 10) - 2 = 0, and not write to A	2	5	18
12	T1 write A	T1 write 0(from Time = 11) to A	0	5	18
13	T1 read B	T1 read 5 from B	0	5	18
14	T1: B = B + 1	T1: B = 5(from Time = 13) + 1 = 6, and not write to B	0	5	18
15	T1 write B	T1 write 6(B at Time = 14) to B	0	6	18
16	T1 commit	T1 commit	0	6	18

So, after time step 16, A = 0, B = 6, and product = 18