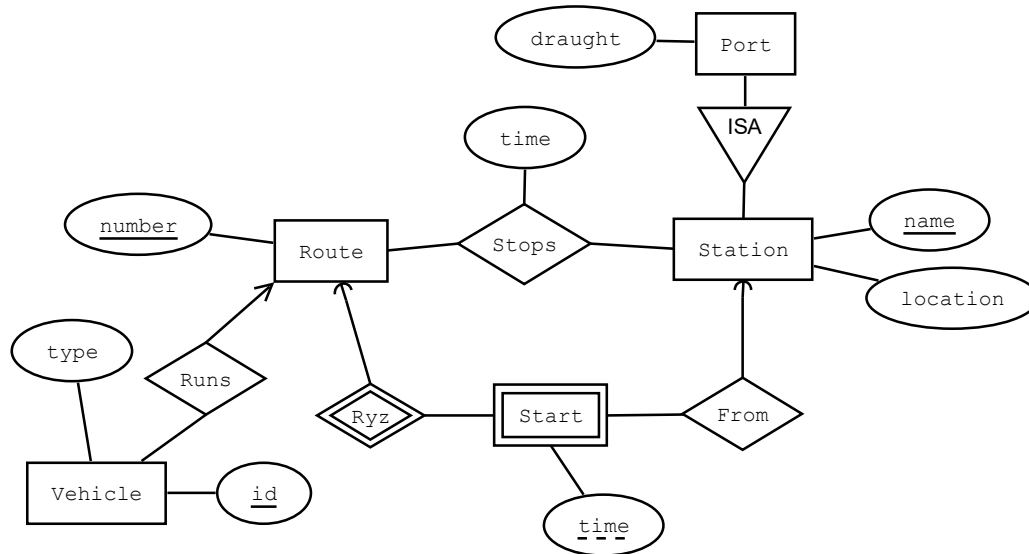


1. a)



b)

$X(x_1, x_2)$

$Z(z_1, z_2)$

$Y(y_1, y_2, z)$

$z \rightarrow Z.z_1$

$Rxyz(x, z, y_1, y_2, yz)$

$x \rightarrow X.x_1$

$z \rightarrow Z.z_1$

$(y_1, y_2, yz) \rightarrow Y.(y_1, y_2, z)$

2

a) Update anomaly e.g. when changing the name of Hermione in on of the two places it occurs.

Deletion anomaly e.g. if deleting all grades issued for course TDA357 and losing the information on the number of credits the course has.

b)

student\_id -> student\_name

course\_id -> credits

(student\_id, course\_id) -> grade

c)

Students(student\_id, student\_name) -- 4 rows

Courses(course\_id, credits) -- 3 rows

Grades(student\_id, course\_id, grade) -- 6 rows

3) SQL Queries (not actually tested these, but they sure look pretty...)

a)

```
SELECT username, email, contents
FROM users
WHERE receiver='admin' AND sender=username AND readtime IS NULL
ORDER BY sendtime ASC
```

b)

```
SELECT AVG(readtime - sendtime)
FROM messages
WHERE readtime IS NOT NULL
```

c)

-- Something like this using WITH, other solutions are possible. Comments are not needed.

WITH

-- Number of messages sent from each user to each receiver

```
Cnt AS SELECT sender, receiver, COUNT(*) AS cnt
FROM Messages
GROUP BY sender, receiver
```

,

-- The maximal number for each sender

```
Mx AS SELECT sender, MAX(cnt) as mx FROM cnt
GROUP BY sender
```

,

-- The most common receiver(s) of each sender

```
Common AS SELECT Cnt.sender, receiver FROM Cnt, Mx
WHERE Mx.sender = Cnt.sender AND cnt=mx
```

-- Include users that have not sent any messages

```
SELECT username, receiver
FROM Users LEFT OUTER JOIN Common ON username=sender
```

4

a)

```
 $\pi_{\text{name, quantity}} ($   
   $\sigma_{(\text{designation}='L' \text{ OR } \text{designation}='XL') \text{ AND } \text{size\_number} \leq \text{max\_number} \text{ AND } \text{size\_number} \geq \text{min\_number}} ($   
    Items X Sizes  
  )  
)
```

b)

```
 $\pi_{\text{designation}} ($   
   $\sigma_{\text{min\_number} > \text{max\_number}} ($   
     $(\pi_{\text{designation, min\_number}}(\text{Sizes}))$   
    X  
     $(\pi_{\text{max\_number}}(\sigma_{\text{designation}='M'}(\text{Sizes})))$   
  )  
)
```

c)

```
 $\gamma_{\text{sum(quantity)} \rightarrow \text{total}} ($   
   $\sigma_{(\text{designation}='L' \text{ AND } \text{size\_number} \leq \text{max\_number} \text{ AND } \text{size\_number} \geq \text{min\_number}} ($   
    Items X Sizes  
  )  
)
```

5 Again – haven't tested this, not likely to work out of the box, but good enough for full marks

```
CREATE TABLE Dots_t (  
  x_pos INT,  
  y_pos INT,  
  idnr INT PRIMARY KEY, -- a  
  UNIQUE (x_pos, y_pos) -- a  
);  
  
CREATE Table Conn_t (  
  from_idnr INT REFERENCES Dots_t.idnr -- b  
  ON DELETE CASCADE, -- f  
  to_idnr INT REFERENCES Dots_t.idnr -- b  
  ON DELETE CASCADE, -- f  
  CHECK (from_idnr != to_idnr) -- b  
  PRIMARY KEY (from_idnr, to_idnr) -- b  
);  
  
CREATE View Connections AS  
  SELECT from_idnr, to_idnr FROM Conn_t  
  UNION  
  SELECT to_idnr, from_idnr FROM Conn_t; -- e  
  
CREATE VIEW Dots AS  
  SELECT  
    x_pos,  
    y_pos,  
    idnr,  
    (SELECT COUNT(*) FROM Connections WHERE from_idnr=idnr) AS radix --c  
  FROM Dots_t;  
  
-- Acceptable pseudo-code for trigger:  
CREATE TRIGGER BEFORE INSERT ON Dots_t -- d  
  
IF ((SELECT radix FROM Dots WHERE idnr=NEW.from_idnr) >= 8 OR  
  (SELECT radix FROM Dots WHERE idnr=NEW.to_idnr) >= 8)  
  ROLLBACK;
```

6

a) Here, measurements are simply any objects with a time value, and requests are objects with "user", "request" and "reply" values. One could add a "type" property to distinguish them.

```
[ { "time":1560000000,
  "temperature":21,
  "windspeed":0,
  "rainfall":{"errnum":213, "text":"no value found"}
},
{ "user":"Jonas",
  "request":["temperature","rainfall"],
  "reply":
  { "time":1560000000,
    "temperature":21,
    "rainfall":{"errnum":213, "text":"no value found"}
  }
}
]
```

b)

```
{
  "type":"array",
  "items":{"type":"object",
    "oneOf":[{"$ref":"#/definitions/measurement"},
    {"$ref":"#/definitions/request"}]},
  "definitions":{
    "measurement": { "type":"object",
      "additionalProperties":{"$ref":"#/definitions/value"},
      "required":["time"]},
    "request":{"properties":{"
      "user": { "type":"string"},
      "request": { "type":"array", "items":{"type":"string"}},
      "reply": { "$ref":"#/definitions/measurement"}},
      "required":["user","request","reply"]},
    "value":{"oneOf":[{"type":"integer"}, {"$ref":"#/definitions/error"}]},
    "error":{"type":"object",
      "properties":{" "errnum":{"type":"integer"},
        "text": { "type":"string"}},
      "required":["errnum", "text"]}
  }
}
```

c) Easy points. May be a lot more complicated for other JSON Schemas.

\$.\*.temperature