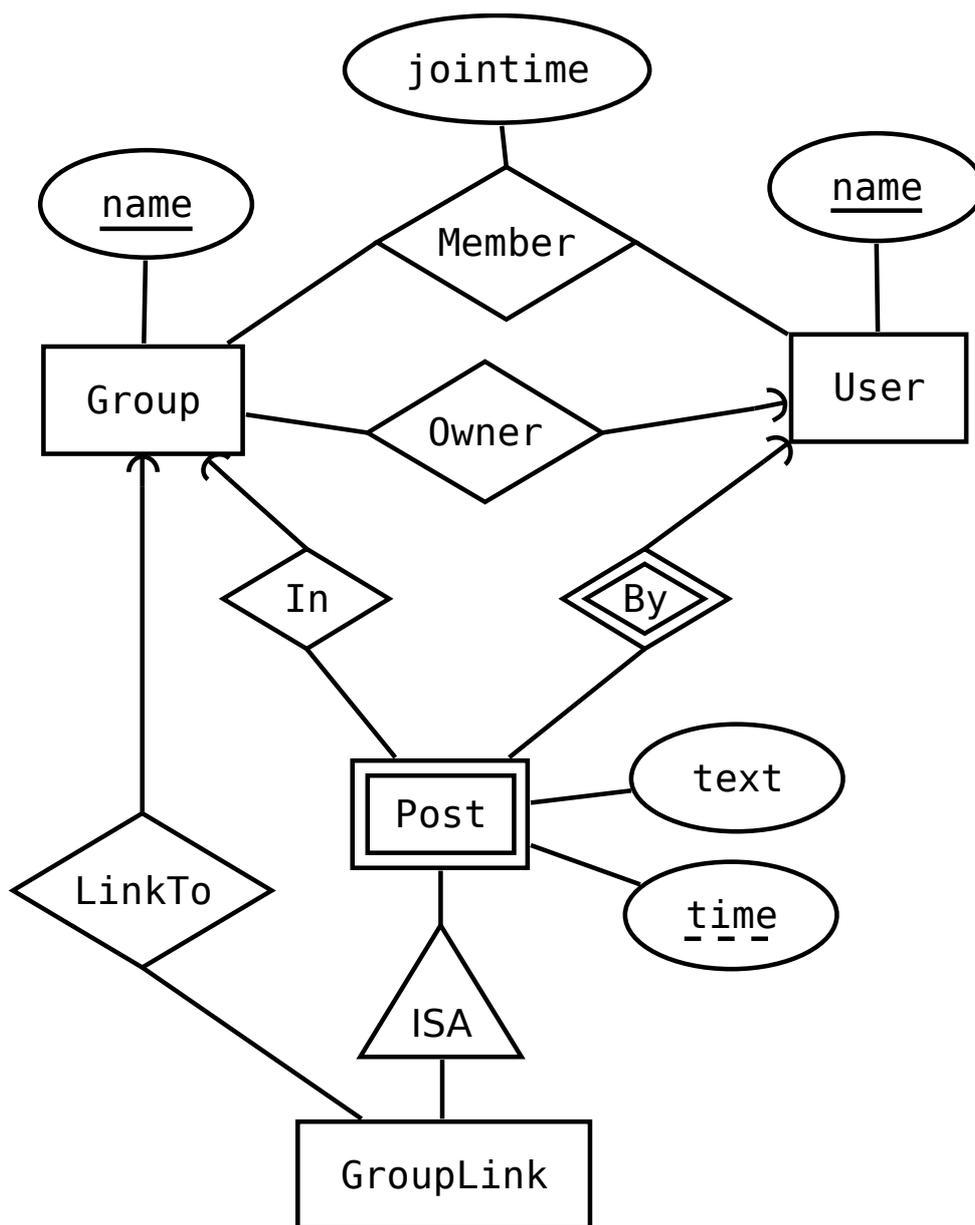


1 a



1 b) valda namn är i stort sett oväsentliga

$X(\underline{x1}, x2)$

$Z(\underline{x1}, z1)$

$x1 \rightarrow X.x1$

$Y(\underline{y1}, \underline{z})$

$z \rightarrow Z.x1$

$R_{zz}(\underline{za}, zb)$

$za \rightarrow Z.x1$

$zb \rightarrow Z.x1$

$R_{xy}(\underline{x}, \underline{y}, \underline{yz}, xy1)$

$x \rightarrow X.x1$

$(y, yz) \rightarrow Y.(y1, z)$

2 a)

$\{E\}^+ = \{E,A,B\}$

$\{B, C\}^+ = \{B,C,E,A\}$

$\{C, D\}^+ = \{C,D\}$

b)

A,C,D

B,C,D

C,D,E

c) One of several possible solutions:

R1(B, A)

R2(E, B)

R3(C, D, E)

3 Untested solutions:

a)

```
SELECT F.x AS x_from, F.y AS y_from, T.x AS x_to, T.y AS y_to, weight
FROM Dots F, Connections, Dots T
WHERE from = F.idnr AND to = T.idnr
```

b) The first operand of the UNION is only to include nodes with RADIX=0, you can still get full points even if you did not include it (providing the rest of your solution is spotless)

```
SELECT idnr, COUNT(weight) AS radix, SUM(weight) AS total
FROM (SELECT idnr, NULL AS weight FROM Connections)
UNION
(SELECT from, weight FROM Connections) UNION
UNION
(SELECT to, weight FROM Connections) AS Combined
GROUP BY idnr;
```

4)

a)

```
 $\pi_{\text{itemname}} ($   
   $\sigma_{\text{value} \geq 1000 \text{ AND level} > 75 \text{ AND id} = \text{item} \text{ AND player} = \text{name}} ($   
    Items X Equipped X Players  
  )  
)
```

b)

```
 $\gamma_{\text{name}, \text{SUM}(\text{value}) \rightarrow \text{total}} ($   
  (PlayerInventory U  $\pi_{\text{item}, \text{player}}$ (Equipped))  
  Xitem=id  
  Items  
)
```

c) First I find unused slots using subtraction (-), same as SQLs EXCEPT/MINUS operator

```
let Unused =  $\sigma_{\text{player} = \text{'jonas'}}$ (  
   $\pi_{\text{player}, \text{equipslot}}$ (PlayerInventory Xid=item Equippable)  
  -  
   $\pi_{\text{player}, \text{equipslot}}$ (Equipped)  
)
```

Then I join the other attributes back in and project the correct ones, here using natural join:

```
R =  $\pi_{\text{id}, \text{equipslot}}$ ( PlayerInventory Xid=item Equippable  $\bowtie$  Unused)
```

5) Untested code

```
CREATE TABLE Warehouses (address TEXT PRIMARY KEY);
```

```
CREATE TABLE Shipment (  
  warehouse TEXT REFERENCES Warehouses, -- d  
  item INT,  
  quantity_change INT,  
  time TIMESTAMP,  
  PRIMARY KEY (warehouse, item, time), -- e  
  CHECK quantity_change != 0 -- f  
);
```

```
CREATE VIEW Inventory AS -- b  
SELECT warehouse, item, SUM(quantity_change) AS quantity  
FROM Shipment  
GROUP BY warehouse, item; -- a
```

-- For c, create a trigger INSTEAD OF UPDATE ON Inventory, that executes this statement:

```
INSERT INTO Shipment  
  VALUES (OLD.warehouse,  
    OLD.item,  
    NEW.quantity-OLD.quantity  
    CURRENT_TIMESTAMP)
```

6)

a)

Any example including user input, concatenating that user input to a query, and an example of malicious input and its consequences. Avoided using prepared statements.

b) There was a small mistake in the schema, saying "file" and "folder" were required instead of filename and foldername.

```
{
  "foldername": "root",
  "contents": [
    {
      "foldername": "temp",
      "contents": [{"filename": "me", "size": 100} ]
    },
    {
      "foldername": "img",
      "contents": [
        {"filename": "pic.jpg", "size": 1234}
      ]
    },
    {"filename": "notes", "size": 100}
  ]
}
```

c)

```
$..[?(@.foldername=="temp")].contents.[*].size
```