

Past 1

Exercise 2.2.1

- a) The attributes of The relation Accounts are acctNo, type, and balance. And, the attributes of The relation customers are firstName, lastName, idNo, and account.
- b) The tuples of The relation Accounts are as following;

1	12345	Saving	12000
2	23456	checking	1000
3	34567	Savings	25

And, the tuples of The relation customers are as following;

1	Robbie	Banks	901-222	12345
2	Lena	Hard	805-333	12345
3	Lena	Hard	805-333	23456

- c) From the relation Accounts

12345	Savings	12000
↓ acctNo	↓ type	↓ balance
And the relation	customers	
↑ firstName	↑ lastName	↑ idNo
Robbie	Banks	901-222
		↑ account
		12345

- d) For The relation Accounts the relation schema is Bank-Accounts (acctNo, type, balance), and the relation schema for The relation customers is Customer's-Accounts (firstName, lastName, idNo, account).
- e) For The relation Accounts the database schema could be Bank-Accounts (acctNo, type, balance). And, the database schema for customer's Account could be Customer's-Account (idNo, accountNo, type, balance).

4) A suitable domain for the relation Accounts could be

acctNo	type	balance
Integer	String	Integer

And, a suitable domain for the relation Customers could be;

FirstName	lastName	idNo	account
String	String	Integer	Integer

5) Another equivalent way to present each relation is;

<u>acctNo</u>	<u>balance</u>	<u>type</u>
12345	12000	Savings

And,

<u>FirstName</u>	<u>lastName</u>	<u>account</u>	<u>idNo</u>
Robbie	Banks	12345	901-222

Exercise 2.2.2

Additional examples that could serve as keys of relations are ID numbers, phone numbers, IP addresses, and so on.

Exercise 2.2.3

- There are 36 ways to represent a relation instance if the instance has three attributes and three tuples.
- There are 2880 ways to represent a relation instance if the instance has four attributes and five tuples.
- There are $n! \times m!$ ways to represent a relation instance if the instance has n attributes and m tuples.

Part 2

Exercise 2.3.1

- A suitable schema for relation product could be

```
CREATE TABLE product (model CHAR(25) NOT NULL,  
                        model CHAR(15) PRIMARY KEY,  
                        type CHAR(25) NOT NULL);
```

- ```
CREATE TABLE PC (model CHAR(15) PRIMARY KEY,
 speed DECIMAL(4,2) NOT NULL,
 ram INTEGER NOT NULL,
 hd INTEGER NOT NULL,
 price DECIMAL(7,2) NOT NULL);
```

- ```
CREATE TABLE Laptop (model CHAR(15) PRIMARY KEY,  
                       speed DECIMAL(4,2) NOT NULL,  
                       ram INTEGER NOT NULL,  
                       hd INTEGER NOT NULL,  
                       screen DECIMAL(3,1) NOT NULL,  
                       price DECIMAL(7,2) NOT NULL);
```

- ```
CREATE TABLE Printer (model CHAR(15) PRIMARY KEY,
 color BOOLEAN NOT NULL,
 type CHAR(10) NOT NULL,
 price DECIMAL(7,2) NOT NULL);
```

e)

ALTER TABLE pointer DROP color;

f)

ALTER TABLE Laptop ADD col CHAR(10) DEFAULT 'none';

### Exercise 2.3.2

a)

CREATE TABLE classes(

class CHAR(25) NOT NULL,

type CHAR(10) NOT NULL,

country CHAR(25) NOT NULL,

numguns INT NOT NULL,

boob DECIMAL(3,1) NOT NULL,

displacement INT NOT NULL,

PRIMARY KEY (class));

b)

CREATE TABLE ships(

name CHAR(25) NOT NULL,

class CHAR(10) NOT NULL,

launched INT NOT NULL);

c)

CREATE TABLE battles(

name CHAR(25) NOT NULL,

date INT NOT NULL);

d) CREATE TABLE OUTCOMES (  
SHIP CHAR(25) NOTNULL,  
BATTLE CHAR(10) NOTNULL,  
RESULT BOOLEAN NOTNULL);

e)

ALTER TABLE CLASSES DROP bote;

f)

ALTER TABLE SHIPS ADD YARD CHAR(10) DEFAULT 'none';

### Exercise 2.4.1

a)  $R1 := \sigma_{\text{speed} > 3.0}(\text{PC})$

b) we select makes, model from product where model  
in (select model from laptop where hd > 100);

c) (select model, price from PC where model in  
(select model from product where makes = 'B'))  
union (select model, price from laptop where model  
in (select model from product where makes = 'B'))  
union (select model, price from printer where model  
in (select model from product where makes = 'B'))

d) we select model from printer where typ = 'lazer'  
and color = 'true'.

e) we select DISTINCT makes from product p where  
p.type = 'Laptop' EXCEPT select DISTINCT makes  
from product p where p.type = 'PC'.

f)

$$PC1 := p(PC)$$

$$PC2 := p(PC)$$

$$PCCross := PC1 \times PC2$$

$$R1 := \sigma_{PC1.hd = PC2.hd \text{ AND } PC1.model > PC2.model} (PCCross)$$

$$R2 := \pi_{PC1.hd} (R1)$$

### Exercise 2.4.3

a)

$$\pi_{class, country} (\sigma_{base \geq 16} (classes))$$

b)

$$\pi_{name} (\sigma_{launched < 1921} (classes \bowtie ships))$$

c)

$$\pi_{name} (\sigma_{battle = \text{Denmark Strait AND result} = \text{Sunk}} (outcomes))$$

d)

$$\pi_{name} (\sigma_{launched > 1921 \text{ AND displacement} > 35000} (classes \bowtie ships))$$

e)

$$\pi_{name, displacement, numGuns} (classes (ships \text{ nam} = \text{ship} (battle = \text{'Guadalcanal'} (outcomes))))$$

f)

$$\pi_{name} (ships)$$