

# Midterm 2023 Solutions

## Problem 1:

$$\{A\}^+ = \{A\}$$

$$\{B\}^+ = \{B\}$$

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

$$\{D\}^+ = \{DE\}$$

$$\{E\}^+ = \{E\}$$

$$\{A, B\}^+ = \{A, B, C, D, E\}, \text{ so } AB \text{ is a key}$$

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

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

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

$$\{B, C\}^+ = \{A, B, C, D, E\}, \text{ so } BC \text{ is a key}$$

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

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

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

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

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

$$\{A, B, C\}^+ = \{A, B, C, D, E\} \text{ (superkey)}$$

$$\{A, B, D\}^+ = \{A, B, C, D, E\} \text{ (superkey)}$$

$$\{A, B, E\}^+ = \{A, B, C, D, E\} \text{ (superkey)}$$

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

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

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

$$\{B, C, D\}^+ = \{A, B, C, D, E\} \text{ (superkey)}$$

$$\{B, C, E\}^+ = \{A, B, C, D, E\} \text{ (superkey)}$$

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

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

$$\{A, B, C, D\}^+ = \{A, B, C, D, E\} \text{ (superkey)}$$

$$\{A, B, C, E\}^+ = \{A, B, C, D, E\} \text{ (superkey)}$$

$$\{A, B, D, E\}^+ = \{A, B, C, D, E\} \text{ (superkey)}$$

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

$$\{B, C, D, E\}^+ = \{A, B, C, D, E\} \text{ (superkey)}$$

So, the only keys are AB and BC.

Consider the functional dependencies.

$AB \rightarrow C$  has a key on the right side.  $BC \rightarrow A$  and  $BC \rightarrow D$  have a key on the right side.

Boyce-Codd Normal Form:  $D \rightarrow E$  is a functional dependency, but D is not a key.

Third Normal Form: Since  $D \rightarrow E$  is a functional dependency, but D is not a key, E would have to be atomic. But E is not part of any key. So, this relation is not in Third Normal Form.

Fourth Normal Form: Since a relation in 4NF is in BCNF, and the relation is NOT BCNF, it cannot be in 4NF either.

## Problem 2:

```
CREATE DATABASE IF NOT EXISTS streaming;
USE streaming;
```

```
DROP TABLE IF EXISTS streaming_rate;
DROP TABLE IF EXISTS film;
DROP TABLE IF EXISTS rate;
DROP TABLE IF EXISTS language1;
DROP TABLE IF EXISTS category;
```

```
CREATE TABLE language1 (
    language_id INT PRIMARY KEY,
    name VARCHAR(60) NOT NULL,
    last_update_date DATE NOT NULL
);
```

```
CREATE TABLE category (
    category_id INT PRIMARY KEY,
    name VARCHAR(60) NOT NULL,
    last_update_date DATE NOT NULL
);
```

```
CREATE TABLE rate (
    rate_id INT PRIMARY KEY,
    name VARCHAR(60) NOT NULL,
    currency VARCHAR(60) NOT NULL,
    amount DECIMAL(10 , 2 ) NOT NULL CHECK(amount > 0)
);
```

```
CREATE TABLE film (
    film_id INT PRIMARY KEY,
    title VARCHAR(60) NOT NULL,
    release_year CHAR(4) NOT NULL,
    language_id INT NOT NULL,
    category_id INT NOT NULL,
    last_update_date DATE NOT NULL,
    FOREIGN KEY (language_id)
        REFERENCES language1 (language_id)
        ON UPDATE CASCADE ON DELETE RESTRICT,
    FOREIGN KEY (category_id)
        REFERENCES category (category_id)
        ON UPDATE CASCADE ON DELETE RESTRICT
);
```

```
CREATE TABLE streaming_rate (
    film_id INT,
```

```

location_country VARCHAR(60) NOT NULL,
rate_id INT,
first_day DATE,
last_day DATE,
PRIMARY KEY(film_id, rate_id, first_day),
FOREIGN KEY (film_id)
    REFERENCES film (film_id)
    ON UPDATE CASCADE ON DELETE RESTRICT,
FOREIGN KEY (rate_id)
    REFERENCES rate (rate_id)
    ON UPDATE CASCADE ON DELETE RESTRICT
);

```

### Problem 3:

(a)

```

SELECT DISTINCT
    productScale
FROM
    Products;

```

	productScale
<input checked="" type="checkbox"/>	1:10
<input type="checkbox"/>	1:12
<input type="checkbox"/>	1:18
<input type="checkbox"/>	1:72
<input type="checkbox"/>	1:24
<input type="checkbox"/>	1:32
<input type="checkbox"/>	1:50
<input type="checkbox"/>	1:700

(b)

```

SELECT
    productName, buyPrice, MSRP
FROM
    Products
WHERE
    productScale = '1:10';

```

(c)

```

SELECT
    productName, ROUND(100* MSRP/buyPrice-100,2) AS markup
FROM
    Products
WHERE
    productScale = '1:50';

```

	productName	markup
<input checked="" type="checkbox"/>	1930 Buick Marquette Phaeton	61.27
<input type="checkbox"/>	Diamond T620 Semi-Skirted Tanker	69.50
<input type="checkbox"/>	1962 City of Detroit Streetcar	56.26
<input type="checkbox"/>	2002 Yamaha YZR M1	138.10

```
(d)
SELECT
    productScale, MAX(MSRP)
FROM
    Products
GROUP BY
    productScale;
```

```
(e)
SELECT
    *
FROM
    PRODUCTS
WHERE
    MSRP = (SELECT
            MAX(MSRP)
            FROM
                Products)
;
```

```
(f)
SELECT
    p.productname, p.productScale, p.buyPrice, p.MSRP
FROM
    products p
    JOIN
    (SELECT
        productscale, MAX(MSRP) AS mmsrp
    FROM
        Products
    GROUP BY productscale) tmsrp ON msrp = mmsrp
;
```

	productname	productScale	buyPrice	MSRP
▶	1952 Alpine Renault 1300	1:10	98.58	214.30
	2001 Ferrari Enzo	1:12	95.59	207.80
	1917 Grand Touring Sedan	1:18	86.70	170.00
	P-51-D Mustang	1:72	49.00	84.48
	1962 Volkswagen Microbus	1:24	61.34	127.79
	1974 Ducati 350 Mk3 Desmo	1:32	56.13	102.05
	Diamond T620 Semi-Skirted Tanker	1:50	68.29	115.75
	ATA: B757-300	1:700	59.33	118.65

(g)

```
SELECT
    customers.customerName,
    payments.checkNumber,
    payments.amount,
    payments.paymentDate
FROM
    customers
    JOIN
    payments USING (customerNumber)
WHERE
    payments.amount < 2000;
```

	customerName	checkNumber	amount	paymentDate
▶	Atelier graphique	OM314933	1676.14	2004-12-18
	Baane Mini Imports	FD317790	1491.38	2003-10-28
	La Corne D'abondance, Co.	AD832091	1960.80	2004-09-09
	Marseille Mini Autos	OB648482	1834.56	2005-01-29
	Royale Belge	CC475233	1627.56	2003-04-19
	Royale Belge	MS154481	1128.20	2003-08-22
	Tokyo Collectables, Ltd	JPMR4544	615.45	2005-05-18
	Microscale Inc.	MO743231	1679.92	2004-04-30

(h)

```
SELECT customerName, SUM(amount)
FROM
    payments p INNER JOIN customers c
    USING (customerNumber)
GROUP BY
    p.customerNumber
ORDER BY Sum(amount) DESC
LIMIT 5;
```

	customerName	SUM(amount)
▶	Euro+ Shopping Channel	715738.98
	Mini Gifts Distributors Ltd.	584188.24
	Australian Collectors, Co.	180585.07
	Muscle Machine Inc	177913.95
	Dragon Souvenirs, Ltd.	156251.03

(i)

```
SELECT
    c.customerName, c.country
FROM
    offices o
        INNER JOIN
    employees e USING (officecode)
        INNER JOIN
    customers c ON (c.salesRepEmployeeNumber = e.employeeNumber)
WHERE
    o.country = "USA";
```

(j)

```
SELECT
    c.customerName
FROM
    productLines pl
        INNER JOIN
    products p USING (productLine)
        INNER JOIN
    orderdetails od USING (productcode)
        INNER JOIN
    orders o USING (orderNumber)
        INNER JOIN
    customers c USING (customerNumber)
WHERE
    pl.textDescription LIKE '%replicas%';
```