**Lab 3**

Group 1

Lab team: Ievgenii Nudga, Aryan Sultan

Instructor: Florian Ocker

HAW Hamburg Data Base

**Assignment 8: Functional Dependencies and Normalization**

**A) The (full) Functional Dependencies**

lect\_ID----->Lec\_Name

Prof\_ID---->pro\_name

Note\_ID--->price, quantity

lect\_id,prof\_id ---->note\_id,price,quantity

**B) Primary key for the table**

lect\_id, prof\_id

**C) Transforming the schema to 2NF**

Main Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lec\_id(PK) | Lec\_name | Prof\_id(PK) | Prof-name | Note\_ID | Price | Quantity |
| 24 | DB | 47 | Miller | 5 | 32 | 12 |
| 24 | DB | 272 | Adams | 1 | 35 | 15 |
| 24 | DB | 251 | Meyer | 5 | 32 | 12 |
| 25 | JAVA | 47 | Miller | 3 | 22 | 19 |

1-2NF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lec\_id(FK) | Prof\_id(FK) | Note\_ID(PK) | Price | Quantity |
| 24 | 47 | 5 | 32 | 12 |
| 24 | 272 | 1 | 35 | 15 |
| 24 | 251 | 5 | 32 | 12 |
| 25 | 47 | 3 | 22 | 19 |

2-prof (prof\_id, prof\_name)

|  |  |
| --- | --- |
| Prof\_id(PK) | Prof-name |
| 47 | Miller |
| 272 | Adams |
| 251 | Meyer |
| 47 | Miller |

3-lecture (lect\_id, lec\_name)

|  |  |
| --- | --- |
| Lec\_id(PK) | Lec\_name |
| 24 | DB |
| 24 | DB |
| 24 | DB |
| 25 | JAVA |

*\*\*\*A note for Evgeniy:*

*why we didn’t create a separate table for Noted because*

*Note id need both primary key elements (keep this in mind)*

**D) Transforming the schema to 3NF**

2NF table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lec\_id(FK) | Prof\_id(FK) | Note\_ID(PK) | Price | Quantity |
| 24 | 47 | 5 | 32 | 12 |
| 24 | 272 | 1 | 35 | 15 |
| 24 | 251 | 5 | 32 | 12 |
| 25 | 47 | 3 | 22 | 19 |

**From 2NF to 3NF**

*\*\*\*A note for Evgeniy:*

Here to know price and quantity it is just enough to know Note\_ID and you dot need prof id and lec id.

We bring out this part as a new table

3NF

|  |  |  |
| --- | --- | --- |
| Lec\_id(FK) | Prof\_id(FK) | Note\_ID(FK) |
| 24 | 47 | 5 |
| 24 | 272 | 1 |
| 24 | 251 | 5 |
| 25 | 47 | 3 |

Note

|  |  |  |
| --- | --- | --- |
| Note\_ID(PK) | Price | Quantity |
| 5 | 32 | 12 |
| 1 | 35 | 15 |
| 5 | 32 | 12 |
| 3 | 22 | 19 |

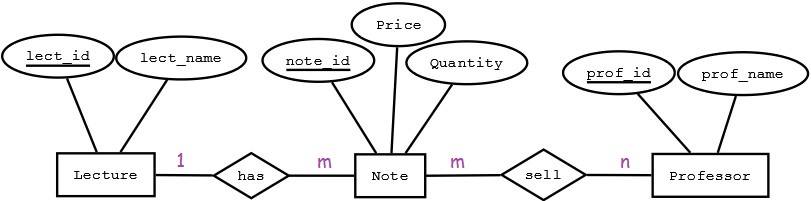
Prof

|  |  |
| --- | --- |
| Prof\_id**(PK)** | Prof-name |
| 47 | Miller |
| 272 | Adams |
| 251 | Meyer |
| 47 | Miller |

Lectures

|  |  |
| --- | --- |
| Lec\_id(PK) | Lec\_name |
| 24 | DB |
| 24 | DB |
| 24 | DB |
| 25 | JAVA |

**E) Draw an ER diagram of the 3NF schema obtained in (D)**



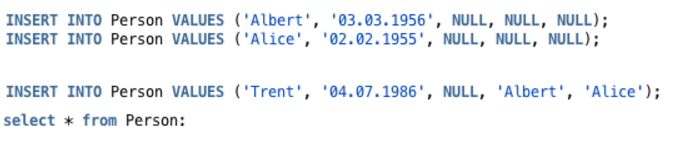
**Assignment 9: DML**

**A)DNA- Analyse**

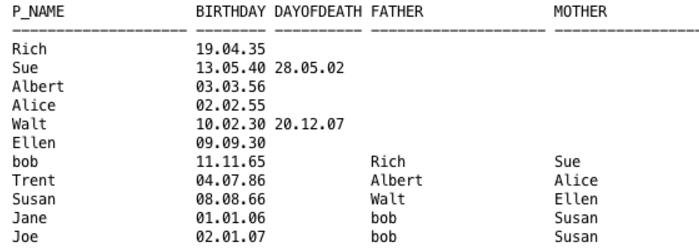
**It should the data of Trent, the true father of Jane and Joe and also his**

**Parents are inserted into the database.**

**The SQL statement is:**



**The output is:**

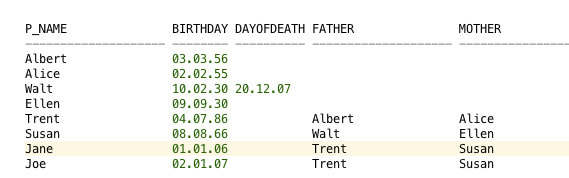


**B) Update**

**As the next SQL statement, Trent will be father of Jane and Joe.**

**The SQL statement is:**

**UPDATE person SET Father = 'Trent' WHERE Father = 'bob';**



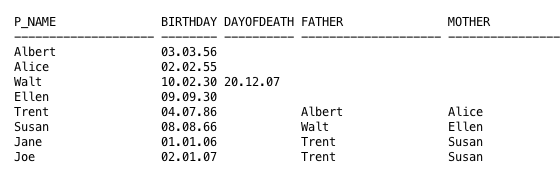
**C) Since the data of Bob and his parents are no longer important to us**

**They are deleted from the table as follow:**

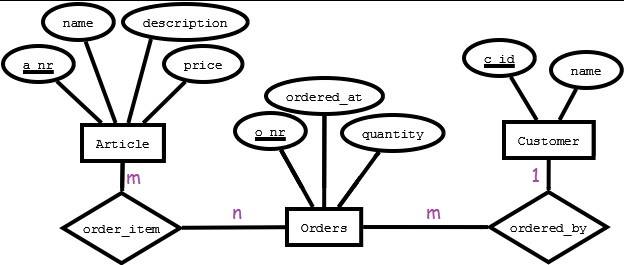
https://lh4.googleusercontent.com/H378AfFNcP-buf73IFui7nOVFmO6PE3kU5NzY57gCDq_ijngi3TxuDQUrAQQkN-JcmskEBcaTgMgGH86gnyylcUc6VvUco4uc-iQnJxawbUmQucxsn7Eviyvztt-8tT173U-CBsS

https://lh4.googleusercontent.com/psA5Ftx0b9hyYKVPxyq61QIyBHwRyCAwo_0JhvFywiWzE0IMliZtRJS8yaWWeai98pnCdClUX09HqMw5T7nW2q9za0r8DAAUnZLJwmQfcNZ72wNVPmHryhweCZmteS84vubr0G1b

**The output is:**

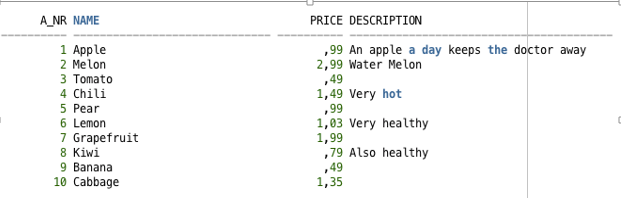


**Assignment 10: Queries**

****

1. **Display all articles**

SELECT \* FROM article;



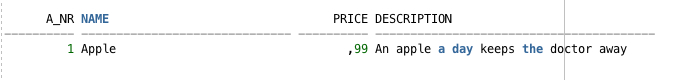
ii. show article with the price 0.99€.

SELECT \* FROM article WHERE price= 0.99;



iii. How much is an “ Apple”?

SELECT \* FROM article WHERE name= 'Apple';



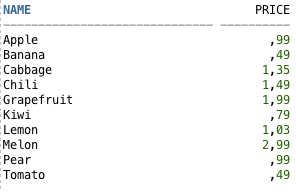
iv. List the name and price of articles without description

SELECT name, price FROM article WHERE description is null;



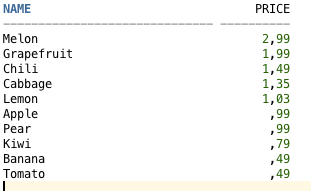
v. List name and articles in alphabetic order.

SELECT name, price FROM ARTICLE order by name ASC , price ASC;-- alphabetic order



vi. List name and articles in price order. Expensive first.

SELECT name, price FROM ARTICLE order by price DESC, name ASC;



vii. What is the average price of articles?

SELECT AVG(price) FROM article;



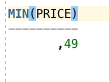
viii. What is the maxumiun price of articles?

SELECT MAX(price) FROM article;



ix. What is the lowest price?

SELECT MIN(price) FROM article;



x. list the order numbers of Ringo

SELECT o\_nr FROM customer ,

orders  WHERE customer.name = 'Ringo' AND customer.c\_id = orders.c\_id ;



xi. List are coustomers and coustomers numbers.

SELECT customer.name, Orders.o\_nr FROM customer

JOIN Orders

ON customer.c\_Id = Orders.c\_Id;



xii. Liste all customer numbers ( also those who not orders sth) and order numbers

SELECT customer.name, Orders.o\_nr FROM customer

LEFT JOIN Orders

ON customer.c\_Id = Orders.c\_Id;



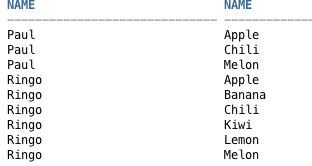
xiii. What are the articles ever ordered by every single customer? Give the customers' and articles' names. Eliminate duplicates. Sort the output alphabetically for both names.

SELECT distinct customer.name,Articel.name from customer

JOIN orders NR  ON customer.c\_id= NR.c\_id

JOIN order\_item Nr\_Item ON Nr\_Item.o\_nr= NR.o\_nr

JOIN article Articel ON Articel.a\_nr= Nr\_Item.a\_nr ORDER BY customer.name, Articel.name;



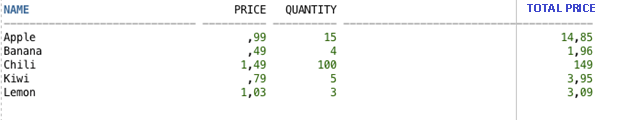
xiv. Display an summary for order 5: For every order item, give the article name, the article's price per unit, the quantity, and the price of the order item. Sort the output alphabetically according to the articles' names.

SELECT article.name, article.price,order\_item.quantity,

order\_item.quantity\*article.price total price

FROM order\_item JOIN article ON order\_item.a\_nr=article.a\_nr

WHERE order\_item.o\_nr=5 ORDER BY article.name;



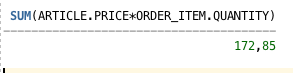
xv. How expensive is the whole order with o\_nr 5?

SELECT SUM(article .price\*order\_item.quantity)

FROM order\_item

JOIN article

ON article.a\_nr = order\_item.a\_nr WHERE order\_item.o\_nr = 5;



xvi. Give a summary of all orders: display the o\_nr and the total price. HINT: use "group by"

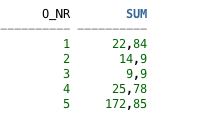
SELECT order\_item.o\_nr, SUM(article .price\*order\_item.quantity)AS SUM

FROM order\_item

JOIN article ON article.a\_nr = order\_item.a\_nr

JOIN ORDERS ON orders.O\_nr = order\_item.O\_nr

GROUP BY order\_item.o\_nr;



xvii. Give a summary of all orders: display the o\_nr, the customer name, and the total price. HINT: use "group by".

SELECT order\_item.o\_nr, customer.name, SUM(article.price\*order\_item.quantity) AS SUM

FROM order\_item

JOIN article ON article.a\_nr = order\_item.a\_nr

JOIN ORDERS ON orders.o\_nr = order\_item.o\_nr

JOIN customer ON orders.c\_id = customer.c\_id

GROUP BY order\_item.o\_nr, customer.name

ORDER BY order\_item.o\_nr;

