ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

BUSINESS ANALYTICS
Master of Science

Athens University of Economics and Business

School of Business

Department of Management Science & Technology

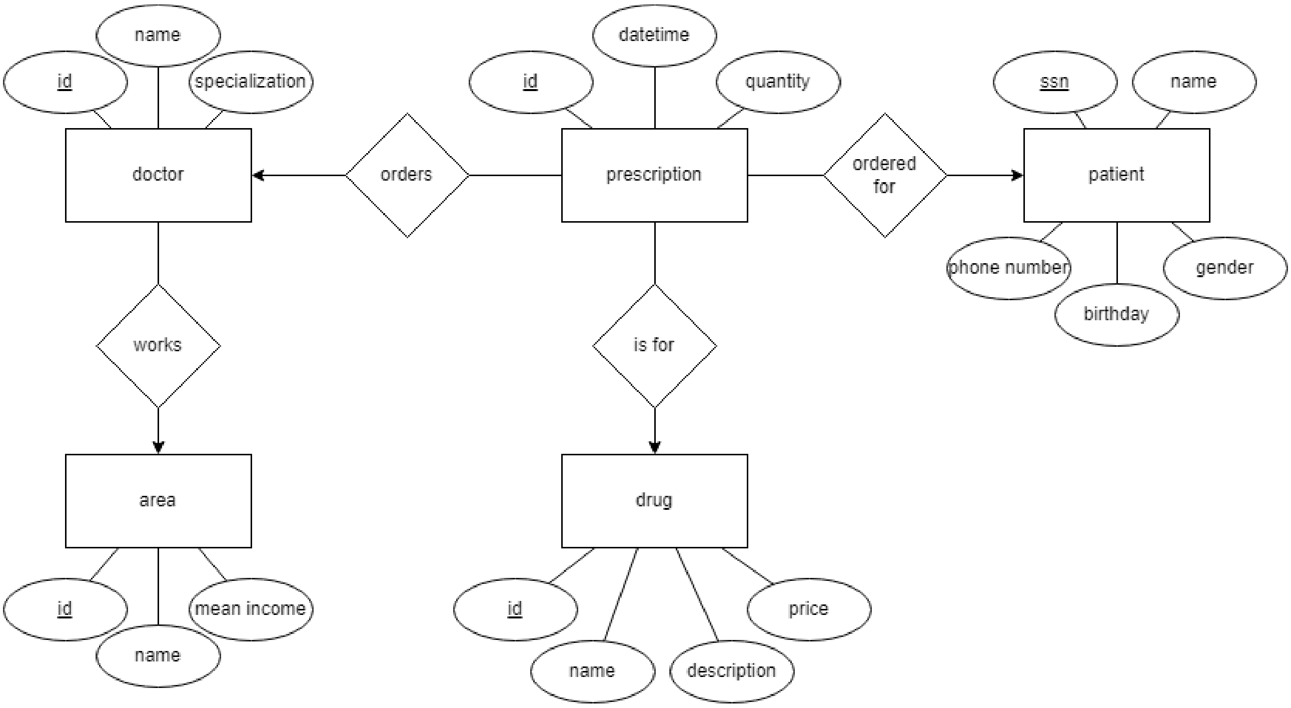
Master of Science in Business Analytics

Program:	Full-time
Quarter:	1st (Fall Quarter)
Course:	Data Management and Business Intelligence
Assignment №:	1
Students (Registration №):	Gkouma Konstantina (f2822208), Souflas Eleftherios-Efthymios (f2822217)

Data Management & Business Intelligence
Assignment #1

Students: Gkouma Konstantina, Souflas Eleftherios-Efthymios

1. Entity-Relationship Diagram (ERD).



2. Relational Schema in MySQL.

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

CREATE TABLE `area` (
  `idarea` int NOT NULL AUTO_INCREMENT,
  `name` varchar(45) NOT NULL,
  `mean_income` decimal(15,2) NOT NULL,
  PRIMARY KEY (`idarea`)
);

CREATE TABLE `doctor` (
  `iddoctor` int NOT NULL AUTO_INCREMENT,
  `name` varchar(45) NOT NULL,
  `specialization` varchar(45) NOT NULL,
  `idarea` int NOT NULL,
  PRIMARY KEY (`iddoctor`),
  UNIQUE KEY `iddoctor_UNIQUE` (`iddoctor`),
  KEY `doctor_area_idx` (`idarea`),
  CONSTRAINT `doctor_area` FOREIGN KEY (`idarea`) REFERENCES `area`
(`idarea`)
);
```

```
CREATE TABLE `drug` (
  `iddrug` int NOT NULL AUTO_INCREMENT,
  `name` varchar(45) NOT NULL,
  `description` varchar(45) DEFAULT NULL,
  `price` decimal(15,2) NOT NULL,
  PRIMARY KEY (`iddrug`)
);

CREATE TABLE `patient` (
  `ssn_patient` char(11) NOT NULL,
  `name` varchar(45) NOT NULL,
  `phone_number` varchar(45) DEFAULT NULL,
  `birthday` date NOT NULL,
  `gender` enum('male','female') NOT NULL,
  PRIMARY KEY (`ssn_patient`),
  UNIQUE KEY `ssn_UNIQUE` (`ssn_patient`)
);

CREATE TABLE `prescription` (
  `idprescription` int NOT NULL AUTO_INCREMENT,
  `datetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `quantity` int NOT NULL,
  `iddoctor` int NOT NULL,
  `ssn_patient` char(11) NOT NULL,
  `iddrug` int NOT NULL,
  PRIMARY KEY (`idprescription`),
  UNIQUE KEY `idprescription_UNIQUE` (`idprescription`),
  KEY `prescr_doctor_idx` (`iddoctor`),
  KEY `prescr_patient_idx` (`ssn_patient`),
  KEY `prescr_drug_idx` (`iddrug`),
  CONSTRAINT `prescr_doctor` FOREIGN KEY (`iddoctor`) REFERENCES `doctor`
(`iddoctor`),
  CONSTRAINT `prescr_drug` FOREIGN KEY (`iddrug`) REFERENCES `drug`
(`iddrug`),
  CONSTRAINT `prescr_patient` FOREIGN KEY (`ssn_patient`) REFERENCES
`patient` (`ssn_patient`) ON UPDATE CASCADE
);
```

3. SQL Queries.

```
-- Query (a)
SELECT
    ssn_patient 'SSN', name 'Name'
FROM
    patient a
WHERE
    DATEDIFF(SYSDATE(), birthday) / 365 >= 30
        AND UPPER(gender) = 'MALE'
        AND EXISTS( SELECT
            1
        FROM
            prescription b
        WHERE
            a.ssn_patient = b.ssn_patient
                AND YEAR(datetime) = 2021);

-- Query (b)
SELECT
    ssn_patient AS 'SSN'
FROM
    patient a
WHERE
    UPPER(gender) = 'FEMALE'
        AND EXISTS( SELECT
            1
        FROM
            prescription b,
            drug c
        WHERE
            a.ssn_patient = b.ssn_patient
                AND b.iddrug = c.iddrug
                AND YEAR(datetime) = 2021
        GROUP BY ssn_patient
        HAVING SUM(b.quantity * c.price) > 1000);

-- Query (c)
SELECT
    a.idarea AS 'Area ID',
    a.name AS 'Area Name',
    IFNULL(SUM(c.quantity * d.price), 0) AS 'Total Amount of Drugs'
FROM
    area a
        LEFT JOIN
    doctor b ON (a.idarea = b.idarea)
        LEFT JOIN
    prescription c ON (b.iddoctor = c.iddoctor)
        LEFT JOIN
    drug d ON (c.iddrug = d.iddrug)
GROUP BY a.idarea;

-- Query (d)
SELECT
    iddrug AS 'Drug ID',
    SUM(IF(Month = 1, Total, 0)) AS 'Jan 2021',
    SUM(IF(Month = 2, Total, 0)) AS 'Feb 2021',
    SUM(IF(Month = 3, Total, 0)) AS 'Mar 2021',
    SUM(IF(Month = 4, Total, 0)) AS 'Apr 2021',
    SUM(IF(Month = 5, Total, 0)) AS 'May 2021',
    SUM(IF(Month = 6, Total, 0)) AS 'Jun 2021',
    SUM(IF(Month = 7, Total, 0)) AS 'Jul 2021',
    SUM(IF(Month = 8, Total, 0)) AS 'Aug 2021',
    SUM(IF(Month = 9, Total, 0)) AS 'Sep 2021',
    SUM(IF(Month = 10, Total, 0)) AS 'Oct 2021',
    SUM(IF(Month = 11, Total, 0)) AS 'Nov 2021',
    SUM(IF(Month = 12, Total, 0)) AS 'Dec 2021'
```

```
FROM
    (SELECT
        a.iddrug AS 'Drug',
            MONTH(datetime) AS 'Month',
            IFNULL(SUM(a.quantity * b.price), 0) AS 'Total'
    FROM
        prescription a, drug b
    WHERE
        YEAR(datetime) = 2021
            AND a.iddrug = b.iddrug
    GROUP BY a.iddrug , MONTH(datetime)) a
        RIGHT JOIN
    drug b ON (a.Drug = b.iddrug)
GROUP BY iddrug;

-- Query (e)
SELECT
    a.iddoctor AS 'Doctor ID',
    a.name AS 'Doctor Name',
    IFNULL(SUM(b.quantity * c.price), 0) AS 'Total Amount of
Prescriptions'
FROM
    doctor a
        LEFT JOIN
    prescription b ON (a.iddoctor = b.iddoctor)
        LEFT JOIN
    drug c ON (b.iddrug = c.iddrug)
WHERE
    a.iddoctor IN (SELECT
            iddoctor
        FROM
            doctor a,
            area b
        WHERE
            a.idarea = b.idarea
                AND mean_income BETWEEN 20000 AND 30000)
GROUP BY a.iddoctor;

-- Query (f)
SELECT
    specialization AS 'Specialization',
    COUNT(idprescription) AS 'Total Number of Prescriptions'
FROM
    doctor a
        LEFT JOIN
    (SELECT
        *
    FROM
        prescription
    WHERE
        YEAR(datetime) = 2021) b ON (a.iddoctor = b.iddoctor)
GROUP BY specialization;
```

```
-- Query (g)
WITH prescr2020 AS (
SELECT
    a.iddrug, IFNULL(SUM(c.quantity * a.price), 0) AS total
FROM
    drug a
        LEFT JOIN
    (SELECT
        *
    FROM
        prescription
    WHERE
        YEAR(datetime) = 2020) c ON (a.iddrug = c.iddrug)
GROUP BY a.iddrug
),
prescr2021 AS (
SELECT
    a.iddrug, IFNULL(SUM(b.quantity * a.price), 0) AS total
FROM
    drug a
        LEFT JOIN
    (SELECT
        *
    FROM
        prescription
    WHERE
        YEAR(datetime) = 2021) b ON (a.iddrug = b.iddrug)
GROUP BY a.iddrug
)
SELECT
    a.iddrug AS 'Drug ID',
    CASE
        WHEN 100 * (b.total - a.total) / a.total IS NOT NULL THEN
ROUND(100 * (b.total - a.total) / a.total, 1)
        WHEN a.total = 0 AND b.total = 0 THEN 0
        ELSE CONCAT('Undefined (from 0 to ', b.total, ')')
    END AS 'Percentage Change'
FROM
    prescr2020 a,
    prescr2021 b
WHERE
    a.iddrug = b.iddrug;

-- Query (h)
WITH male AS (
SELECT
    a.iddrug, IFNULL(SUM(b.quantity * a.price), 0) AS total
FROM
    drug a
        LEFT JOIN
    prescription b ON (a.iddrug = b.iddrug)
        LEFT JOIN
    patient c ON (b.ssn_patient = c.ssn_patient)
WHERE
    UPPER(gender) = 'MALE'
        AND YEAR(datetime) = 2021
GROUP BY a.iddrug
),
female AS (
SELECT
    a.iddrug, IFNULL(SUM(b.quantity * a.price), 0) AS total
FROM
    drug a
        LEFT JOIN
    prescription b ON (a.iddrug = b.iddrug)
        LEFT JOIN
    patient c ON (b.ssn_patient = c.ssn_patient)
WHERE
    UPPER(gender) = 'FEMALE'
        AND YEAR(datetime) = 2021
GROUP BY a.iddrug
)
SELECT
    a.iddrug AS 'Drug ID',
    IFNULL(b.total, 0) AS Male,
    IFNULL(c.total, 0) AS Female
FROM
    drug a
        LEFT JOIN
    male b ON (a.iddrug = b.iddrug)
        LEFT JOIN
    female c ON (a.iddrug = c.iddrug)
GROUP BY a.iddrug;
```

4. Python Code to connect to Database and execute query.

```
# Use first pip install to download and install "MySQL Connector" driver
# cmd> python -m pip install mysql-connector-python
import mysql.connector

# Create the connection to the database
mydb = mysql.connector.connect(
    host='127.0.0.1',
    user='my_username',
    password='my_password'
)

# Query
myCursor = mydb.cursor()
myCursor.execute('SELECT a.`idprescription` AS "ID",'
                 'DATE_FORMAT(a.`datetime`, "%d/%m/%Y %H:%i:%S") AS "Date
& Time",'
                 'CONCAT(b.`name`, " (tel: ", b.`phone_number`, ")") AS
"Patient",'
                 'CONCAT(c.`name`, " (",c.`specialization`, ")") AS
"Doctor",'
                 'd.`name` AS "Drug",'
                 'cast(d.`price` as char(50)) AS "Price",'
                 'a.`quantity` AS "Quantity" '
                 'FROM `prescriptions`.`prescription` a,'
                 '`prescriptions`.`patient` b,'
                 '`prescriptions`.`doctor` c,'
                 '`prescriptions`.`drug` d '
                 'WHERE a.`ssn_patient` = b.`ssn_patient` '
                 'AND a.`iddoctor` = c.`iddoctor` '
                 'AND a.`iddrug` = d.iddrug')

# Print query's column names
print(myCursor.column_names)
# Alternatively
# numFields = len(myCursor.description)
# fieldNames = [i[0] for i in myCursor.description]
# print(fieldNames)

# Print query's result
myResult = myCursor.fetchall()
for i in myResult:
    print(i)
```