Lab 5 CS254

Name: Abhiraj Mengade
Roll no: 201cs102

Q1. Build a basic database(of your choice) and explore the usage of following string function:

```
CREATE TABLE students (
id INT PRIMARY KEY,
first_name VARCHAR(255) NOT NULL,
last_name VARCHAR(255),
email VARCHAR(255),
gender ENUM('M', 'F'),
city VARCHAR(255) NOT NULL,
country VARCHAR(255) NOT NULL
);
INSERT INTO students VALUES
(1, 'Alvy', 'Pietruszka', 'apietruszka0@toplist.cz', 'M', 'Paris 15',
'France'),
(2, 'Maryl', 'Collacombe', 'mcollacombe1@ycombinator.com', 'F', 'Tarbes',
'France'),
(3, 'Amos', 'Skiplorne', 'askiplorne2@unicef.org', 'M', 'Tarbes',
'France'),
(4, 'Verla', 'Brabham', 'vbrabham3@forbes.com', 'F', 'Poitiers',
'France'),
(5, 'Devi', 'Staynes', 'dstaynes4@google.es', 'F', 'Angers', 'France'),
(6, 'Rikki', 'Egleton', 'regleton5@opera.com', 'F', 'Soissons', 'France'),
(7, 'Rosy', 'Dragonette', 'rdragonette6@netlog.com', 'F', 'Saskatoon',
'Canada'),
(8, 'Jethro', 'Creek', 'jcreek7@google.fr', 'M', 'Nanterre', 'France'),
(9, 'Wye', 'Alliott', 'walliott8@youtu.be', 'M', 'Orlando', 'United
States'),
(10, 'Tait', 'Asplin', 'tasplin9@hao123.com', 'M', 'Hénin-Beaumont',
'France');
```

```sql
CREATE TABLE students (
id INT PRIMARY KEY,
first_name VARCHAR(255) NOT NULL,
last_name VARCHAR(255),
email VARCHAR(255),
gender ENUM('M', 'F'),
city VARCHAR(255) NOT NULL,
country VARCHAR(255) NOT NULL
);

INSERT INTO students VALUES
(1, 'Alvy', 'Pietruszka', 'apietruszka0@toplist.cz', 'M', 'Paris 15', 'France'),
(2, 'Maryl', 'Collacombe', 'mcollacombe1@ycombinator.com', 'F', 'Tarbes', 'France'),
(3, 'Amos', 'Skiplorne', 'askiplorne2@unicef.org', 'M', 'Tarbes', 'France'),
(4, 'Verla', 'Brabham', 'vbrabham3@forbes.com', 'F', 'Poitiers', 'France'),
(5, 'Devi', 'Staynes', 'dstaynes4@google.es', 'F', 'Angers', 'France'),
(6, 'Rikki', 'Egleton', 'regleton5@opera.com', 'F', 'Soissons', 'France'),
(7, 'Rosy', 'Dragonette', 'rdragonette6@netlog.com', 'F', 'Saskatoon', 'Canada'),
(8, 'Jethro', 'Creek', 'jcreek7@google.fr', 'M', 'Nanterre', 'France'),
(9, 'Wye', 'Alliott', 'walliott8@youtu.be', 'M', 'Orlando', 'United States'),
(10, 'Tait', 'Asplin', 'tasplin9@hao123.com', 'M', 'Hénin-Beaumont', 'France');

SELECT * FROM students;
```

STDIN

Input for the program (Optional)

Output:

id	first_name	last_name	email	gender
1	Alvy	Pietruszka	apietruszka0@toplist.cz	
2	Maryl	Collacombe	mcollacombe1@ycombinator	
3	Amos	Skiplorne	askiplorne2@unicef.org	
4	Verla	Brabham	vbrabham3@forbes.com	F
5	Devi	Staynes	dstaynes4@google.es	F
6	Rikki	Egleton	regleton5@opera.com	F
7	Rosy	Dragonette	rdragonette6@netlog.com	
8	Jethro	Creek	jcreek7@google.fr	M
9	Wye	Alliott	walliott8@youtu.be	M
10	Tait	Asplin	tasplin9@hao123.com	M

a.) CHAR_LENGTH()

```sql
SELECT id, first_name, CHAR_LENGTH(first_name) AS first_name_length
FROM students;
```

```sql
SELECT id, first_name, CHAR_LENGTH(first_name) AS first_name_length
FROM students;
```

STDIN

Input for the program (Optional)

Output:

id	first_name	first_name_length
1	Alvy	4
2	Maryl	5
3	Amos	4
4	Verla	5
5	Devi	4
6	Rikki	5
7	Rosy	4
8	Jethro	6
9	Wye	3
10	Tait	4

b.) CONCAT()

```sql
SELECT CONCAT(first_name, ' ', last_name) AS name
FROM students;
```

```
SELECT CONCAT(first_name, ' ', last_name) AS name
FROM students;
```

STDIN

Input for the program (Optional)

Output:

```
name
Alvy Pietruszka
Maryl Collacombe
Amos Skiplorne
Verla Brabham
Devi Staynes
Rikki Egleton
Rosy Dragonette
Jethro Creek
Wye Alliott
Tait Asplin
```

c.) INSERT()

```
SELECT INSERT(first_name, 1, 2, '##') AS masked_name
FROM students;
```

```
SELECT INSERT(first_name, 1, 2, '##') AS masked_name
FROM students;
```

STDIN

Input for the program (Optional)

Output:

```
masked_name
##vy
##ryl
##os
##rla
##vi
##kki
##sy
##thro
##e
##it
```

d.)LCASE()

```
SELECT LOWER(first_name) AS first_name_in_lower_case
FROM students;
```

```
SELECT LOWER(first_name) AS first_name_in_lower_case
FROM students;
```

STDIN

Input for the program (Optional)

Output:

```
first_name_in_lower_case
alvy
maryl
amos
verla
devi
rikki
rosy
jethro
wye
tait
```

e.) LENGTH()

```
SELECT first_name, LENGTH(first_name) AS `LENGTH(first_name)`
FROM students;
```

```
SELECT first_name, LENGTH(first_name) AS `LENGTH(first_name)`
FROM students;
```

Input for the program (Optional)

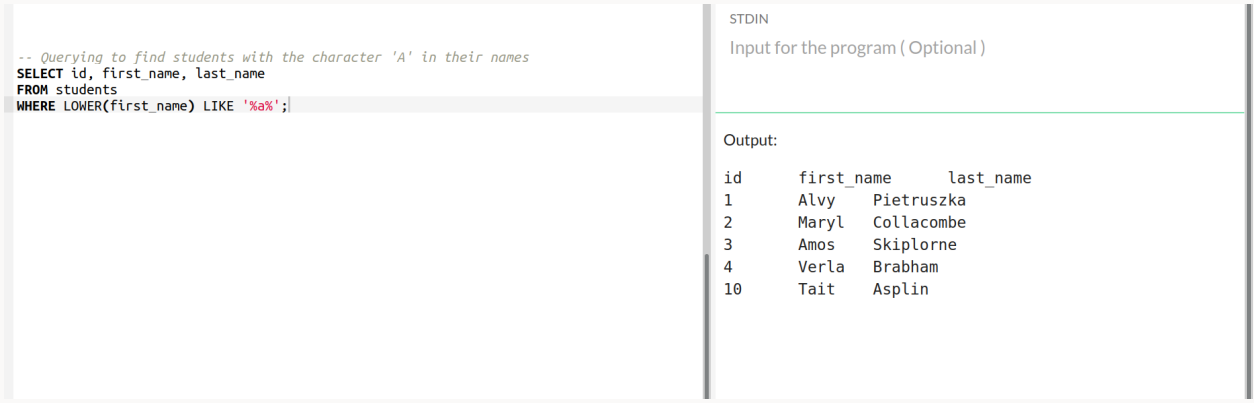
Output:

```
first_name      LENGTH(first_name)
Alvy    4
Maryl   5
Amos    4
Verla   5
Devi    4
Rikki   5
Rosy    4
Jethro  6
Wye     3
Tait    4
```

f.) LIKE

```
-- Querying to find students with the character 'A' in their names
SELECT id, first_name, last_name
```

```
FROM students
WHERE LOWER(first_name) LIKE '%a%';
```



```
-- Querying to find students with the character 'A' in their names
SELECT id, first_name, last_name
FROM students
WHERE LOWER(first_name) LIKE '%a%';
```

STDIN

Input for the program (Optional)

Output:

```
id      first_name      last_name
1       Alvy    Pietruszka
2       Maryl   Collacombe
3       Amos    Skiplorne
4       Verla   Brabham
10      Tait    Asplin
```

g.) TRIM()

```
SELECT CONCAT('', first_name, '')
AS padded_name, TRIM(CONCAT('', first_name,' ' ))
AS trimmed_name
FROM students;
```

```
SELECT CONCAT('', first_name, '')
AS padded_name, TRIM(CONCAT('', first_name,' ' ))
AS trimmed_name
FROM students;
```

STDIN

Input for the program (Optional)

Output:

```
padded_name     trimmed_name
Alvy    Alvy
Maryl   Maryl
Amos    Amos
Verla   Verla
Devi    Devi
Rikki   Rikki
Rosy    Rosy
Jethro  Jethro
Wye     Wye
Tait    Tait
```

h.) STRCMP()

```
SELECT first_name, STRCMP(first_name, 'Alvy') AS is_Alvy
FROM students;
```

```
SELECT first_name, STRCMP(first_name, 'Alvy') AS is_Alvy
FROM students;
```

Input for the program (Optional)

Output:

```
first_name      is_Alvy
Alvy    0
Maryl   1
Amos    1
Verla   1
Devi    1
Rikki   1
Rosy    1
Jethro  1
Wye     1
Tait    1
```

i.) SUBSTR()

```
SELECT first_name, STRCMP(first_name, 'Alvy') AS is_Alvy
FROM students;
```

```
SELECT first_name, SUBSTR(email, 1, 10) AS `SUBSTR(email, 1, 10)`
FROM students;
```

STDIN

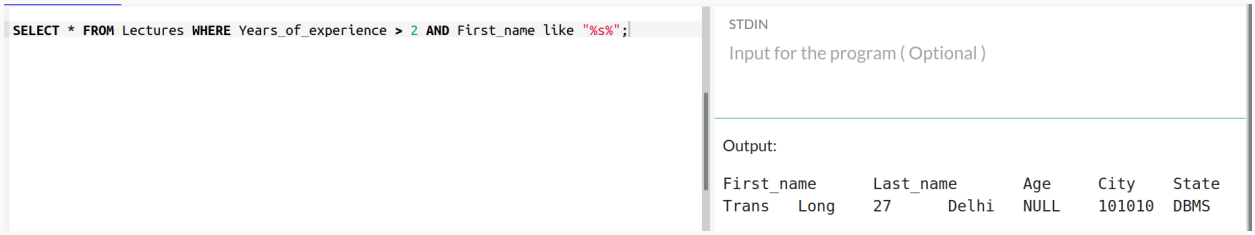
Input for the program (Optional)

Output:

```
first_name      SUBSTR(email, 1, 10)
Alvy    apietruszk
Maryl   mcollacomb
Amos    askiplorne
Verla   vbrabham3@
Devi    dstaynes4@
Rikki   regleton5@
Rosy    rdragonett
Jethro  jcreek7@go
Wye     walliott8@
Tait    tasplin9@h
```

j.) SUBSTR()

```sql
SELECT first_name, SUBSTR(email, 1, 10) AS `SUBSTR(email, 1, 10)`
FROM students;
```



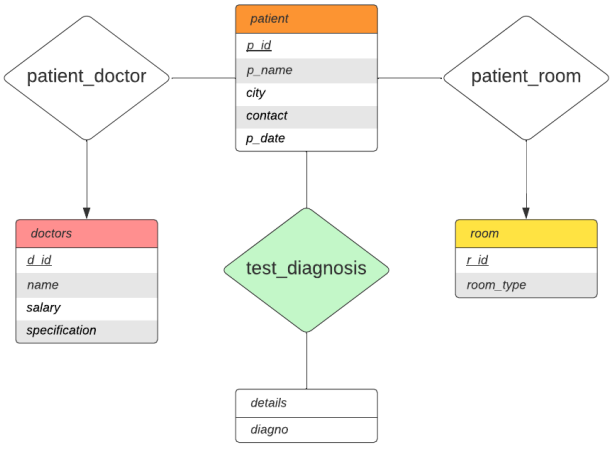
Q2. Create database with PATIENT (p_id, r_id, d_id, p_name, city ,contact, p_date), DOCTORS(d_id, name, salary , specification) ROOM(r_id, room_type),TEST & DIAGNOSIS(p_id, diagno, diag_details).

```sql
CREATE TABLE room (
r_id INT PRIMARY KEY,
room_type ENUM('Single', 'Double', 'Deluxe')
);
CREATE TABLE doctors (
d_id INT PRIMARY KEY,
name VARCHAR(255) NOT NULL,
salary NUMERIC,
specification VARCHAR(255)
);
CREATE TABLE patient (
p_id INT PRIMARY KEY,
r_id INT NOT NULL,
d_id INT,
p_name VARCHAR(255) NOT NULL,
city VARCHAR(255),
contact VARCHAR(25),
```

```
p_date DATE NOT NULL,
FOREIGN KEY (d_id) REFERENCES doctors(d_id),
FOREIGN KEY (r_id) REFERENCES room(r_id)
);
CREATE TABLE test_diagnosis (
p_id INT NOT NULL,
diagno VARCHAR(255),
details VARCHAR(255),
FOREIGN KEY (p_id) REFERENCES patient(p_id)
);

INSERT INTO room VALUES
(101, 'Double'),
(102, 'Deluxe'),
(103, 'Deluxe'),
(104, 'Double'),
(105, 'Double'),
(106, 'Deluxe'),
(107, 'Deluxe'),
(108, 'Single'),
(109, 'Double'),
(110, 'Single');

INSERT INTO doctors VALUES
(1, 'Peria Hawe', 27180, 'ENT'),
(2, 'Cecelia Pietranek', 46110, 'Neurologist'),
(3, 'Dayle Halladay', 30770, 'Pediatrician'),
(4, 'Timothee Britnell', 86150, 'Cardiologist'),
(5, 'Delcina Poundsford', 58610, 'Neurologist'),
(6, 'Darsie Worling', 50180, 'Pediatrician'),
(7, 'Salmon Hunnicot', 70560, 'Pediatrician'),
(8, 'Don Deery', 55700, 'General Practitioner'),
(9, 'Bail Lascell', 63860, 'Cardiologist'),
(10, 'Isacco Eccleshare', 59860, 'Pediatrician');

INSERT INTO patient VALUES
(1, 101, 3, 'Ernest Earney', 'Arroyo Seco', '+54 (552) 637-9873',
'2021-11-09'),
(2, 107, 9, 'Madalyn Braunstein', 'Huddinge', '+46 (360) 362-8676',
'2021-10-13'),
```

```
(3, 103, 2, 'Miguel Juggings', 'Cilongkrangpusaka', '+62 (957) 331-6191',
'2021-05-19'),
(4, 101, 7, 'Bride Sussams', 'Monkey Hill', '+1 (764) 512-5023',
'2015-07-27'),
(5, 105, 2, 'Ettie Cubin', 'Maxixe', '+258 (870) 777-0419', '2016-12-22'),
(6, 109, 1, 'Francesca Gunn', 'Jednorożec', '+48 (645) 932-5694',
'2021-09-11'),
(7, 102, 5, 'Onida Petto', 'Huangtan', '+86 (797) 140-7353',
'2021-09-26'),
(8, 109, 10, 'Brinn Brabbins', 'Tshikapa', '+242 (263) 609-4294',
'2021-10-30'),
(9, 106, 2, 'Dulcia Beauman', 'Maglaj', '+387 (988) 292-9263',
'2015-12-21'),
(10, 105, 7, 'Smith Aleksich', 'Guocun', '+86 (525) 561-1589',
'2016-02-25');

INSERT INTO test_diagnosis VALUES
(1, 'Biopsy', 'OK'),
(1, 'Clotting', 'OK'),
(1, 'Imaging', 'OK'),
(4, 'LP', 'OK'),
(5, 'Imaging', 'Fatty deposits'),
(5, 'Clotting', 'OK'),
(7, 'Ultrasound', 'OK'),
(8, 'LP', 'Bacterial Meningitis'),
(6, 'Clotting', 'OK'),
(10, 'Endoscopy', 'OK');
```

ER Diagram:



a.) List the patient details with multiple diagnosis records.

```
SELECT p.p_id, p.p_name, COUNT(*) AS NUM_tests
FROM patient p, test_diagnosis td
WHERE p.p_id=td.p_id
GROUP BY p.p_id
HAVING num_tests>1
```

```
SELECT p.p_id, p.p_name, COUNT(*) AS NUM_tests
FROM patient p, test_diagnosis td
WHERE p.p_id=td.p_id
GROUP BY p.p_id
HAVING num_tests>1
```

STDIN

Input for the program (Optional)

Output:

```
p_id    p_name  NUM_tests
1       Ernest Earney   3
5       Ettie Cubin     2
```

c.) Fetch the doctors who do not have any patients.

```
SELECT *
FROM doctors
WHERE d_id NOT IN (
SELECT d_id FROM patient
);
```

```
SELECT *
FROM doctors
WHERE d_id NOT IN (
SELECT d_id FROM patient
);
```

STDIN

Input for the program (Optional)

Output:

```
d_id    name    salary  specification
4       Timothee Britnell       86150   Cardiologist
6       Darsie Worling  50180   Pediatrician
8       Don Deery       55700   General Practitioner
```

d.) Display doctors salary in ascending order

```
SELECT * FROM doctors
ORDER BY salary;
```

```
SELECT * FROM doctors
ORDER BY salary;
```

STDIN

Input for the program (Optional)

Output:

```
d_id    name    salary  specification
1       Peria Hawe      27180   ENT
3       Dayle Halladay  30770   Pediatrician
2       Cecelia Pietranek       46110   Neurologist
6       Darsie Worling  50180   Pediatrician
8       Don Deery       55700   General Practitioner
5       Delcina Poundsford      58610   Neurologist
10      Isacco Eccleshare       59860   Pediatrician
9       Bail Lascell    63860   Cardiologist
7       Salmon Hunnicot 70560   Pediatrician
4       Timothee Britnell       86150   Cardiologist
```

e.) Display the each patient details through diagd_details

```
FROM test_diagnosis td JOIN patient p
ON td.p_id = p.p_id;
```

f.) Display the number of patients for each doctor
. Only include doctors with more than 3 patients.

```
-- Querying for >=3 patients to avoid empty set
SELECT d.d_id, d.name, COUNT(*) AS num_patients
FROM patient p, doctors d
WHERE p.d_id = d.d_id
GROUP BY d.d_id
HAVING num_patients >= 3;
```

```
-- Querying for >=3 patients to avoid empty set
SELECT d.d_id, d.name, COUNT(*) AS num_patients
FROM patient p, doctors d
WHERE p.d_id = d.d_id
GROUP BY d.d_id
HAVING num_patients >= 3;
```

Output:

```
d_id    name    num_patients
2       Cecelia Pietranek       3
```

g.) Display the doctors who are treating patients from r_id 102 to 105.

```
SELECT d.d_id, d.name AS doctor, p.p_name, r.r_id
```

```
FROM doctors d, patient p, room r
WHERE
p.d_id = d.d_id AND
p.r_id = r.r_id AND
r.r_id BETWEEN 102 AND 105;
```

```
(10, 'Endoscopy', 'OK');

SELECT d.d_id, d.name AS doctor, p.p_name, r.r_id
FROM doctors d, patient p, room r
WHERE
p.d_id = d.d_id AND
p.r_id = r.r_id AND
r.r_id BETWEEN 102 AND 105;
```

Output:

d_id	doctor	p_name	r_id
5	Delcina Poundsford	Onida Petto	102
2	Cecelia Pietranek	Miguel Juggings	103
2	Cecelia Pietranek	Ettie Cubin	105
7	Salmon Hunnicot Smith	Aleksich	105

h.) Display the patients details according to their joining dates

```
SELECT * FROM patient
ORDER BY p_date;
```

```
SELECT * FROM patient
ORDER BY p_date;
```

Output:

p_id	r_id	d_id	p_name	city	contact	p_date
4	101	7	Bride Sussams	Monkey Hill		
9	106	2	Dulcia Beauman	Maglaj	+387 (98	
10	105	7	Smith Aleksich	Guocun	+86 (525	
5	105	2	Ettie Cubin	Maxixe	+258 (87	
3	103	2	Miguel Juggings	Cilongkrangpusak		
6	109	1	Francesca Gunn	Jednorożec		
7	102	5	Onida Petto	Huangtan		
2	107	9	Madalyn Braunstein	Huddinge		
8	109	10	Brinn Brabbins	Tshikapa		
1	101	3	Ernest Earney	Arroyo Seco		

i.) Count the patients who took deluxe rooms

```
SELECT COUNT(*) AS deluxe_occupants
FROM patient p, room r
WHERE p.r_id = r.r_id AND r.room_type = 'Deluxe';
```

```
SELECT COUNT(*) AS deluxe_occupants
FROM patient p, room r
WHERE p.r_id = r.r_id AND r.room_type = 'Deluxe';
```

Output:

```
deluxe_occupants
4
```

j.) Display name of the doctor with salary less than 40000

```
SELECT * FROM doctors
WHERE salary < 40000;
```

```
SELECT * FROM doctors
WHERE salary < 40000;
```

Output:

```
d_id    name    salary  specification
1       Peria Hawe      27180   ENT
3       Dayle Halladay  30770   Pediatrician
```

k.) Display the patients joined before 10.10.2017.

```
SELECT * FROM patient
WHERE p_date < '2017-10-10';
```

```
SELECT * FROM patient
WHERE p_date < '2017-10-10';
```

Output:

p_id	r_id	d_id	p_name	city	contact	p_date
4	101	7	Bride	Sussams	Monkey Hill	
5	105	2	Ettie	Cubin	Maxixe	+258 (87
9	106	2	Dulcia	Beauman	Maglaj	+387 (98
10	105	7	Smith	Aleksich	Guocun	+86 (525

Q3. Create database for below Schema:

(Add 10 entries for each table)

- BOOK(Book_id, Title, Publisher_Name, Pub_date)

- BOOK_AUTHORS(Book_id,Author_Name)

- PUBLISHER(FName,LName,Address, Phone)

- BOOK_COPIES(Book_id, Programme_id, No-of_Copies)

- BOOK_LENDING(Book_id, Programme_id, Card_No, Date_Out, Due_Date)

```
CREATE TABLE book (
book_id INT PRIMARY KEY,
title VARCHAR(255) NOT NULL,
publisher_name VARCHAR(255),
pub_date DATE
);
CREATE TABLE book_authors (
book_id INT NOT NULL,
author_name VARCHAR(255),
FOREIGN KEY (book_id) REFERENCES book(book_id)
);
```

```sql
CREATE TABLE book_copies (
book_id INT NOT NULL,
programme_id INT PRIMARY KEY,
num_copies INT NOT NULL,
FOREIGN KEY (book_id) REFERENCES book(book_id)
);
CREATE TABLE publisher (
f_name VARCHAR(80),
l_name VARCHAR(80),
address VARCHAR(255),
phone VARCHAR(25)
);
CREATE TABLE book_lending (
book_id INT NOT NULL,
programme_id INT NOT NULL,
card_no INT,
date_out DATE NOT NULL,
due_date DATE,
FOREIGN KEY (book_id) REFERENCES book(book_id),
FOREIGN KEY (programme_id) REFERENCES book_copies(programme_id)
);


INSERT INTO book VALUES
(1, 'Don Quixote of La Mancha', 'Henry Munchan', '2021-6-6'),
(2, 'Crime and Punishment', 'Nicoli Ducker', '2021-5-7'),
(3, 'Dracula', 'Verena Lesper', '2022-1-23'),
(4, 'Great Expectations', 'Verena Lesper', '2021-6-20'),
(5, 'David Copperfield', 'Nicoli Ducker', '2021-10-27'),
(6, 'Divine Comedy', 'Danella Stallan', '2021-10-21'),
(7, 'Canterbury Tales', 'Danella Stallan', '2021-8-22'),
(8, 'In Search of Lost Time', 'Danella Stallan', '2021-7-18'),
(9, 'Treasure Island', 'Danella Stallan', '2021-11-24'),
(10, 'Lolita', 'Jaclin Baynam', '1998-3-20');


INSERT INTO book_authors VALUES
(1, "Leo Tolstoy" ),
(2, "Fyodor Doestevisky"),
(3, "Charles Dickens"),
```

```
(4, "Charles Dickens"),
(5, "Charles Dickens"),
(6, "Dante"),
(7, "Alexandre Dumas"),
(8, "Marcel Proust"),
(9, "Robert Louis Stevenson"),
(10, "Robert Louis Stevenson");

INSERT INTO book_copies VALUES
(2, 2, 44),
(1, 1, 87),
(3, 3, 25),
(4, 4, 57),
(5, 5, 100),
(6, 6, 24),
(7, 7, 100),
(8, 8, 66),
(9, 9, 12),
(10, 10, 55);

INSERT INTO publisher VALUES
('Verena', 'Lesper', '788 Johnson Center', '+86-766-117-9865'),
('Nicoli', 'Ducker', '77 Meadow Vale Hill', '+62-829-382-1965'),
('Danella', 'Stallan', '45206 Loomis Place', '+387-102-305-5492'),
('Luigi', 'Bareham', '605 Steensland Plaza', '+86-357-334-8465'),
('Jaclin', 'Baynam', '50 Rigney Place', '+62-808-723-7265');


INSERT INTO book_lending VALUES
(1, 1, 32, '2017-02-1', '2017-03-1'),
(3, 3, 31, '2017-02-5', '2017-03-5'),
(2, 2, 1, '2018-07-6', '2018-08-6'),
(3, 3, 5, '2021-09-20', '2021-10-20'),
(4, 4, 5, '2020-09-20', '2020-10-20');
```

a.).Retrieve details of all books in the library – id, title, name of publisher

```
SELECT book_id, title, publisher_name FROM book;
```

.

```
SELECT book_id, title, publisher_name FROM book;
```

STDIN

Input for the program (Optional)

Output:

```
book_id title   publisher_name
1       Don Quixote of La Mancha        Henry Munchan
2       Crime and Punishment    Nicoli Ducker
3       Dracula Verena Lesper
4       Great Expectations      Verena Lesper
5       David Copperfield       Nicoli Ducker
6       Divine Comedy   Danella Stallan
7       Canterbury Tales        Danella Stallan
8       In Search of Lost Time  Danella Stallan
9       Treasure Island Danella Stallan
10      Lolita  Jaclin Baynam
```

b.).Retrieve the books which have been borrowed from Jan 2017 to March 2017.

```
SELECT b.book_id, b.title, bl.date_out
FROM book b, book_lending bl
WHERE
b.book_id = bl.book_id AND
bl.date_out BETWEEN '2017-01-01' AND '2017-03-31';
```

```
SELECT b.book_id, b.title, bl.date_out
FROM book b, book_lending bl
WHERE
b.book_id = bl.book_id AND
bl.date_out BETWEEN '2017-01-01' AND '2017-03-31';
```

STDIN

Input for the program (Optional)

Output:

```
book_id title   date_out
1       Don Quixote of La Mancha        2017-02-01
3       Dracula 2017-02-05
```

c.).Delete a book in the BOOK table. Update the contents of other tables to reflect this data manipulation operation.

```
DELETE FROM book_authors WHERE book_id = 1;
DELETE FROM book_lending WHERE book_id = 1;
DELETE FROM book_copies WHERE book_id = 1;
```

```
DELETE FROM book WHERE book_id = 1;
```

```
DELETE FROM book_authors WHERE book_id = 1;
DELETE FROM book_lending WHERE book_id = 1;
DELETE FROM book_copies WHERE book_id = 1;
DELETE FROM book WHERE book_id = 1;

SELECT * FROM book;
```

STDIN

Input for the program (Optional)

Output:

```
book_id title   publisher_name  pub_date
2       Crime and Punishment    Nicoli Ducker   2021-05-
3       Dracula Verena Lesper   2022-01-23
4       Great Expectations      Verena Lesper   2021-06-
5       David Copperfield       Nicoli Ducker   2021-10-
6       Divine Comedy   Danella Stallan 2021-10-21
7       Canterbury Tales        Danella Stallan 2021-08-
8       In Search of Lost Time  Danella Stallan 2021-07-
9       Treasure Island Danella Stallan 2021-11-24
10      Lolita  Jaclin Baynam   1998-03-20
```

d.).Retrieve the details of the books(id, title, publisher name, year) published on the date 20-03-1998.

```
SELECT book_id, title, publisher_name, YEAR(pub_date) AS pub_year
FROM book
WHERE pub_date = '1998-03-20';
```

```
SELECT book_id, title, publisher_name, YEAR(pub_date) AS pub_year
FROM book
WHERE pub_date = '1998-03-20';
```

Input for the program (Optional)

Output:

```
book_id title   publisher_name  pub_year
10      Lolita  Jaclin Baynam   1998
```

e.).Retrieve the books published by a particular author

```
SELECT b.title FROM
book b, book_authors ba
WHERE b.book_id = ba.book_id AND ba.author_name = 'Charles Dickens';
```

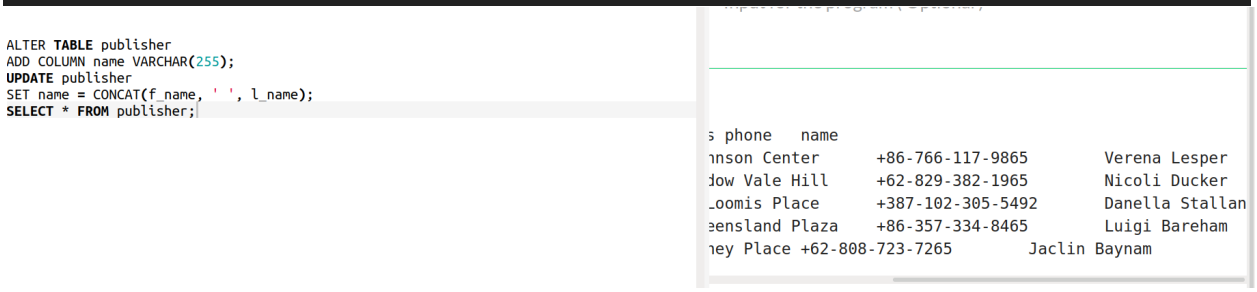
```
SELECT b.title FROM
book b, book_authors ba
WHERE b.book_id = ba.book_id AND ba.author_name = 'Charles Dickens';
```

Output:

```
title
Dracula
Great Expectations
David Copperfield
```

f.).Create a new column ‘name’ in the Publishers table. Combine FName and LName and print it in column name.

```
ALTER TABLE publisher
ADD COLUMN name VARCHAR(255);
UPDATE publisher
SET name = CONCAT(f_name, ' ', l_name);
```



g.).Write a query to display the first day of the month (in datetime format) two months before the current month from the date of publication of the book “DBMS”.

```
SELECT DATE_FORMAT(DATE_ADD(pub_date, INTERVAL -2 MONTH), '%Y-%m-01')
FROM book
WHERE title = 'Dracula';
```

```
SELECT DATE_FORMAT(DATE_ADD(pub_date, INTERVAL -2 MONTH), '%Y-%m-01')
FROM book
WHERE title = 'Dracula';
```

Output:

```
DATE_FORMAT(DATE_ADD(pub_date, INTERVAL -2 MONTH), '%Y-%
2021-11-01
```

h.).Write a query to get the years in which more than 3 books were published.

```
SELECT YEAR(pub_date) AS year, COUNT(*) AS num_books
FROM book
GROUP BY year
HAVING num_books > 3;
```

```
SELECT YEAR(pub_date) AS year, COUNT(*) AS num_books
FROM book
GROUP BY year
HAVING num_books > 3;
```

Output:

```
year    num_books
2021    8
```

i.).Print the number of copies of a particular book.

```
SELECT book_id, SUM(num_copies) AS total_copies
FROM book_copies
GROUP BY book_id
HAVING book_id = 3;
```

```
SELECT book_id, SUM(num_copies) AS total_copies
FROM book_copies
GROUP BY book_id
HAVING book_id = 3;
```

Output:

```
book_id total_copies
3       25
```