Exercise 1

```
--Creating tables
CREATE TABLE IF NOT EXISTS Parts
(
        PID INT NOT NULL,
        Pname varchar(30),
        Color VARCHAR(30),
  PRIMARY KEY (PID)
);

CREATE TABLE IF NOT EXISTS Suppliers
(
        SID INT NOT NULL,
        Sname VARCHAR(50),
        Address VARCHAR(50),
   PRIMARY KEY (SID)
);

CREATE TABLE IF NOT EXISTS "Catalog"
(
        SID INT,
        PID INT,
        "COST" DECIMAL(5, 2),
        FOREIGN KEY (SID) REFERENCES Suppliers(SID),
        FOREIGN KEY (PID) REFERENCES Parts(PID)
);

--Filling tables

INSERT INTO suppliers (sid, sname, address)
VALUES (1, 'Yosemite Sham', 'Devil's canyon, AZ'),
(2, 'Wiley E. Coyote', 'RR Asylum, NV'),
(3, 'Elmer Fudd', 'Carrot Patch, MN');

INSERT INTO parts(pid, pname, color) VALUES
(1, 'Red1', 'Red'),
(2, 'Red1', 'Red'),
(3, 'Green1', 'Green'),
(4, 'Blue1', 'Blue'),
(5, 'Red3', 'Red');

INSERT INTO "Catalog" (sid, pid, "COST")
VALUES (1, 1, 10),
(1, 2, 20),
(1, 3, 30),
(1, 4, 40),
(1, 5, 50),
(2, 1, 9),
(2, 3, 34),
(2, 5, 48);
```

1. **Find the names of suppliers who supply some red part.**

```
SELECT DISTINCT S.sname
FROM suppliers S, parts Pa, "Catalog" Ca
WHERE Pa.color = 'Red' AND Ca.pid = Pa.pid and Ca.sid = S.sid;
```

sname
Wiley E. Coyote
Yosemite Sham

2. **Find the sids of suppliers who supply some red or green part.**

```
SELECT DISTINCT CA.sid
FROM parts Pa, "Catalog" Ca
WHERE (Pa.color = 'Red' OR Pa.color = 'Green') AND Ca.pid = Pa.pid;
```

sid
1
2

3. **Find the sids of suppliers who supply some red part or are at 221 Packer Street**

```
SELECT S.sid
FROM suppliers S
WHERE S.address = ' 221 Packer Street' OR S.sid IN (
  SELECT Ca.sid
  From "Catalog" Ca, parts Pa
  WHERE Pa.color = 'Red' and Pa.pid = Ca.pid
  );
```

sid
1
2

4. **Find the sids of suppliers who supply every red or green part.**

```
SELECT DISTINCT Ca.sid
FROM "Catalog" Ca
WHERE NOT EXISTS (
  Select Pa.pid
  From parts Pa
  Where (Pa.color = 'Red' OR Pa.color = 'Green') AND NOT EXISTS (
    SELECT "Catalog".sid
    FROM "Catalog"
    WHERE Ca.sid = "Catalog".sid AND Ca.pid = "Catalog".pid
    )
  );
```

sid
1
2

5. **Find the sids of suppliers who supply every red part or supply every green part.**

```
SELECT DISTINCT Ca.sid
FROM "Catalog" Ca
WHERE (NOT EXISTS (
  SELECT P1.pid
  FROM parts P1
  WHERE P1.color = 'Red' AND NOT EXISTS (
    SELECT C1.sid
    FROM "Catalog" C1
    WHERE C1.sid = Ca.sid AND C1.pid= P1.pid
    )
  ))
  OR NOT EXISTS (
    SELECT P2.pid
        FROM parts P2
        WHERE P2.color = 'Green' AND NOT EXISTS (
    SELECT C2.sid
    FROM "Catalog" C2
    WHERE C2.sid = Ca.sid AND C2.pid= P2.pid
    )
  )
```

sid
1
2

6. **Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.**

```
SELECT DISTINCT C1.sid, C2.sid
FROM "Catalog" C1, "Catalog" C2
WHERE C1.pid = C2.pid AND C1.sid != C2.sid AND C1."COST" > C2."COST";
```

sid	sid
1	2
2	1

7. **Find the pids of parts supplied by at least two different suppliers.**

```
SELECT DISTINCT C1.pid
From "Catalog" C1
WHERE EXISTS (
  SELECT C2.sid
  From "Catalog" C2
  Where C1.pid = C2.pid AND C2.sid != C1.sid
  );
```

pid
3
5
1

8. **Find the average cost of the red parts and green parts for each of the suppliers**

```
SELECT DISTINCT Al.sid, ROUND(AVG(Al."COST"),2), Al.color
From ("Catalog" C1 LEFT JOIN parts P1 ON C1.pid = P1.pid) AS Al
WHERE Al.color = 'Red' OR Al.color = 'Green'
Group by Al.sid, Al.color;
```

sid	COST	color
1	26.67	Red
1	30.00	Green
2	28.50	Red
2	34.00	Green

9. **Find the sids of suppliers whose most expensive part costs $50 or more**

```
SELECT Al.sid
FROM (
  SELECT C1.sid, MAX(C1."COST") AS "COST"
  FROM "Catalog" C1
  GROUP BY C1.sid
  ) AS Al
WHERE Al."COST" >= 50
GROUP BY Al.sid;
```

sid
1

Exercise 2

```
--Creating tables
CREATE TABLE author
(
  author_id INT NOT NULL PRIMARY KEY,
  first_name VARCHAR(30),
  last_name VARCHAR(30)
);
CREATE TABLE book
(
  book_id INT NOT NULL PRIMARY KEY,
  book_title VARCHAR(30),
  "month" VARCHAR(30),
  "year" INT,
  editor INT
);
CREATE TABLE pub
(
  pub_id INT NOT NULL PRIMARY KEY,
  title VARCHAR(30),
  book_id INT,
  FOREIGN KEY (book_id) REFERENCES book(book_id)
);
CREATE TABLE another_pub
(
  author_id INT,
  pub_id INT,
  author_position INT,
  FOREIGN KEY (author_id) REFERENCES author(author_id),
  FOREIGN KEY (pub_id) REFERENCES pub(pub_id)
);

--Filling tables
INSERT INTO author (author_id, first_name, last_name)
VALUES (1, 'John', 'McCarthy'),
(2, 'Dennis', 'Ritchie'),
(3, 'Ken', 'Thompson'),
(4, 'Claude', 'Shannon'),
(5, 'Alan', 'Turing'),
(6, 'Alonzo', 'Church'),
(7, 'Perry', 'White'),
(8, 'Moshe', 'Vardi'),
(9, 'Roy', 'Batty');

INSERT iNTO book (book_id, book_title, "month", "year", editor)
VALUES (1, 'CACM', 'April', 1960, 8),
(2, 'CACM', 'July', 1974, 8),
(3, 'BTS', 'July', 1948, 2),
(4, 'MLS', 'November', 1936, 7),
(5, 'Mind', 'October', 1950, NULL),
(6, 'AMS', 'Month', 1941, NULL),
(7, 'AAAI', 'July', 2012, 9),
(8, 'NIPS', 'July', 2012, 9);

INSERT inTO pub (pub_id, title, book_id)
VALUES (1, 'LISP', 1),
(2, 'UNIX', 2),
(3, 'Info Theory', 3),
(4, 'Turing Machines', 4),
(5, 'Turing Test', 5),
(6, 'Lambda Calculus', 6);

INSERT INTO another_pub (author_id, pub_id, author_position)
VALUES (1, 1, 1),
(2, 2, 1),
(3, 2, 1),
```

```
(4, 3, 1),
(5, 4, 1),
(5, 5, 1),
(6, 6, 1);
```

1. • $Author \bowtie_{author_{id}=editor} Book$

```
author INNER JOIN book ON author.author_id = book.editor;
```

But it is better to visualize the result:

```
select *
From author INNER JOIN book ON author.author_id = book.editor;
```

author_id	first_name	last_name	book_id	book_title	month	year	editor
8	Moshe	Vardi	1	CACM	April	1960	8
8	Moshe	Vardi	2	CACM	July	1974	8
2	Dennis	Ritchie	3	BTS	July	1948	2
7	Perry	White	4	MLS	November	1936	7
9	Roy	Batty	7	AAAI	July	2012	9
9	Roy	Batty	8	NIPS	July	2012	9

2. • $\Pi_{first\_name,last\_name}\left(\left(\Pi_{author\_id}(Author) - \Pi_{editor}(Book) \bowtie Author\right)\right)$

```
SELECT DISTINCT author.first_name, author.last_name
From author
where author.author_id IN (
  Select DISTINCT author.author_id
  From author
  where author.author_id NOT IN (
    SELECT Al.book_id
    From (book Cross JOIN author) Al
    )
  );
```

first_name	last_name
Roy	Batty

3. • $\Pi_{author\_id}(Author) - \Pi_{editor}(book)$

```
SELECT author.author_id
FROM author
WHERE author.author_id NOT IN (
  SELECT A1.author_id
  From author A1, book B1
  WHERE A1.author_id = B1.editor
  );
```

author_id
1
3
4
5
6

Exercise 3

1. **• Find the distinct names of all students who score more than 90% in the course numbered 107**

```
SELECT DISTINCT Al.sname
FROM (Courses CROSS JOIN Students) Al
WHERE Al.cid = 107 AND Al.percent >= 90;
```

2. **Find the number of student whose score is 75% or above in each course**

```
SELECT COUNT(Al.sid)
FROM (
        SELECT DISTINCT R.sid
        FROM Registration R
        WHERE MIN(R.percent) >= 75
        GROUP BY R.sid
        ) Al;
```

3. **• Find those students who are registered on no more than 2 courses.**

```
SELECT R.sid
FROM Registration R
where COUNT(R.sid) <= 2
GROUP BY R.sid;
```