SQL:

1. show databases;
2. CREATE TABLE `Client` (ClientId CHAR(36) PRIMARY KEY,FirstName VARCHAR(50) NOT NULL,LastName VARCHAR(50) NOT NULL,BirthDate DATE NULL,Address VARCHAR(256) NULL,City VARCHAR(100) NULL,StateAbbr CHAR(2) NULL,PostalCode VARCHAR(10) NULL,    FOREIGN KEY fk\_Client\_StateAbbr (StateAbbr)        REFERENCES State(StateAbbr));
3. SELECT FirstName, LastName, Address FROM Client;
4. SELECT FirstName, LastName, Address FROM ClientWHERE LastName = "smith";
5. CREATE TABLE Task ( TaskId INT PRIMARY KEY AUTO\_INCREMENT, Title VARCHAR(100) NOT NULL, Details TEXT NULL,  DueDate DATE NOT NULL, EstimatedHours DECIMAL(5, 2) NULL );
6. DROP DATABASE IF EXISTS TrackIt;
7. CREATE DATABASE TrackIt
8. USE TrackIt;
9. CREATE TABLE Worker (WorkerId INT PRIMARY KEY AUTO\_INCREMENT,FirstName VARCHAR(50) NOT NULL,LastName VARCHAR(50) NOT NULL);
10. CREATE TABLE Project (ProjectId CHAR(50) PRIMARY KEY,`Name` VARCHAR(100) NOT NULL,Summary VARCHAR(2000) NULL,DueDate DATE NOT NULL,IsActive BOOL NOT NULL DEFAULT 1);
11. INSERT INTO Worker (WorkerId, FirstName, LastName)  VALUES (1, 'Rosemonde', 'Featherbie');
12. INSERT INTO Worker (FirstName, LastName)  VALUES ('Kingsly', 'Besantie');
13. INSERT INTO Worker (FirstName, LastName) VALUES  
      ('Goldi','Pilipets'),('Dorey','Rulf'),('Panchito','Ashtonhurst');
14. INSERT INTO Worker (WorkerId, FirstName, LastName)  VALUES (50, 'Valentino', 'Newvill');
15. UPDATE Project SET  
      Summary = 'All lessons and exercises for the relational database milestone.',DueDate = '2018-10-15' WHERE ProjectId = 'db-milestone'
16. UPDATE Worker SET LastName = 'Oaks WHERE WorkerId = 2;
17. DELETE FROM Worker WHERE WorkerId = 50;
18. DELETE FROM Worker WHERE FirstName = 'Kingsly';
19. SELECT \*FROM TaskWHERE TaskStatusId BETWEEN 5 AND 8;

Joins:  
 1. SELECT Task.TaskId,Task.Title,TaskStatus.NameFROM TaskStatus INNER JOIN Task ON TaskStatus.TaskStatusId = Task.TaskStatusIdWHERE TaskStatus.IsResolved = 1;

2. SELECT Task.TaskId,Task.Title,TaskStatus.NameFROM TaskStatus INNER JOIN Task ON TaskStatus.TaskStatusId = Task.TaskStatusIdWHERE TaskStatus.IsResolved = 1;

3. SELECT Project.Name, Worker.FirstName, Worker.LastName FROM Project INNER JOIN ProjectWorker ON Project.ProjectId = ProjectWorker.ProjectId INNER JOIN Worker ON ProjectWorker.WorkerId = Worker.WorkerId WHERE Project.ProjectId = 'game-goodboy';

4. SELECT Project.Name,Worker.FirstName,Worker.LastName,Task.TitleFROM ProjectINNER JOIN ProjectWorker ON Project.ProjectId = ProjectWorker.ProjectIdINNER JOIN Worker ON ProjectWorker.WorkerId = Worker.WorkerIdINNER JOIN Task ON ProjectWorker.ProjectId = Task.ProjectId  AND ProjectWorker.WorkerId = Task.WorkerIdWHERE Project.ProjectId = 'game-goodboy';

5. SELECT \*FROM TaskLEFT OUTER JOIN TaskStatus  ON Task.TaskStatusId = TaskStatus.TaskStatusId;

6. SELECT  Project.Name ProjectName, -- An alias makes this more clear.Worker.FirstName,Worker.LastNameFROM ProjectLEFT OUTER JOIN ProjectWorker ON Project.ProjectId = ProjectWorker.ProjectIdLEFT OUTER JOIN Worker ON ProjectWorker.WorkerId = Worker.WorkerId;

7. SELECT CONCAT(w.FirstName, ' ', w.LastName) WorkerName, p.Name ProjectNameFROM Worker w CROSS JOIN Project p WHERE w.WorkerId = 1AND p.ProjectId NOT LIKE 'game-%';

**Aggregate functions:**

**COUNT:**

Counts the number of non-NULL values in a set; works on any non-NULL value

**SUM:**

Sums values in a set; values must be numeric

**AVG:**

Calculates the average of values in a set; values must be numeric

**MIN:**

Determines the minimum value in a set; values must be comparable

**MAX:**

Determines the maximum value in a set; values must be comparable

Count:  
SELECT

COUNT(t.TaskId)

FROM Task t

INNER JOIN TaskStatus s ON t.TaskStatusId = s.TaskStatusId

WHERE s.IsResolved = 1;

**GROUP BY:**

SELECT

IFNULL(s.Name, '[None]') StatusName,

COUNT(t.TaskId) TaskCount

FROM Task t

LEFT OUTER JOIN TaskStatus s ON t.TaskStatusId = s.TaskStatusId

GROUP BY s.Name

ORDER BY s.Name;

**HAVING:**

SELECT

CONCAT(w.FirstName, ' ', w.LastName) WorkerName,

SUM(t.EstimatedHours) TotalHours

FROM Worker w

INNER JOIN ProjectWorker pw ON w.WorkerId = pw.WorkerId

INNER JOIN Task t ON pw.WorkerId = t.WorkerId

AND pw.ProjectId = t.ProjectId

GROUP BY w.WorkerId, w.FirstName, w.LastName;

**SubQueries**:  
SELECT

g.ProjectName,

g.MinTaskId,

t.Title MinTaskTitle

FROM Task t

INNER JOIN (

SELECT

p.Name ProjectName,

MIN(t.TaskId) MinTaskId

FROM Project p

INNER JOIN Task t ON p.ProjectId = t.ProjectId

GROUP BY p.ProjectId, p.Name) g ON t.TaskId = g.MinTaskId;

SELECT

w.FirstName,

w.LastName,

(SELECT COUNT(\*) FROM ProjectWorker

WHERE WorkerId = w.WorkerId) ProjectCount

FROM Worker w;

**Views:**

CREATE VIEW ProjectNameWithMinTaskId AS

SELECT

p.Name ProjectName,

MIN(t.TaskId) MinTaskId

FROM Project p

INNER JOIN Task t ON p.ProjectId = t.ProjectId

GROUP BY p.ProjectId, p.Name;

SELECT \* FROM ProjectNameWithMinTaskId;

SELECT pt.ProjectName, pt.MinTaskId TaskId, t.Title

FROM Task t

INNER JOIN ProjectNameWithMinTaskId pt -- Aliased just like a table.

ON t.TaskId = pt.MinTaskId;