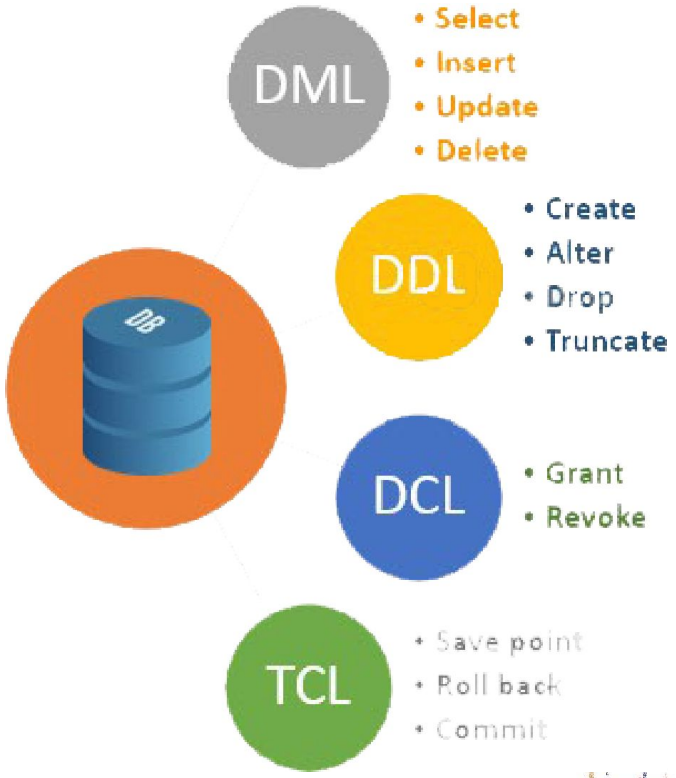
Chapter 5

SQL

SQL

- Structured Query Language
  - Developed at IBM original as SEQUEL
  - Based on Relational Algebra and Tuple Calculus
  - VERY GOOD at handling structured data
- Main components
  - DDL (Data Definition Language)
  - DML (Data Manipulation Language)
    - Select
    - Insert
    - Update
    - Delete



Assumptions

- Working with the same tables/relations

  Sailors(sid:int, sname:string, rating:int, age:real)

  Boats(bid: int, bname:string, color:string)

  Reserves(sid:int, bid: int, day: date)

SELECT Statement

- case INSENSITIVE
- Basic Syntactical Form

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR | IS NULL>] ;
```

Notation:

Blue: keywords

[]: optional arguments

<>: mandatory arguments

|: or

SELECT [DISTINCT] <columns list| *>
FROM <relations list>
[WHERE <conditions| AND| OR>] ;

Simple Queries – all rows

- Get all data from Sailors (all columns – all rows)

```
SELECT *
FROM   Sailors;
```

- With alias for Sailors table

```
SELECT *
FROM   Sailors S;
```

- Get sailor ids and sailor names from Sailors table (all rows)

```
SELECT S.sid, S.sname
FROM   Sailors S;
```

SELECT [DISTINCT] <columns list| *>
FROM <relations list>
[WHERE <conditions| AND| OR>] ;

Simple Queries (2) - Distinct

- Get unique sailor names from Sailors table (all rows)

```
SELECT DISTINCT S.sname
FROM   Sailors S;
```

- Get unique values of sailor names and ratings from Sailors table

```
SELECT DISTINCT S.sname, S.rating
FROM   Sailors S;
```

- WRONG SQL

```
SELECT S.sname DISTINCT S.rating
FROM   Sailors S;
```

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>] ;
```

Simple Queries (3) - tuples

- Get unique sailor names over 18 from Sailors table

```
SELECT DISTINCT S.sname
FROM   Sailors S
WHERE S.age>18;
```

- Get unique sailor names over 18 and rating less or equal 8 from Sailors table

```
SELECT DISTINCT S.sname
FROM   Sailors S
WHERE S.age>18 AND
   S.rating<=8;
```

SELECT [DISTINCT] <columns list| *>
FROM <relations list>
[WHERE <conditions| AND| OR>] ;

Simple Queries (4) conditions

- Get unique sailor names over 18 or rating less or equal 8 from Sailors table

```
SELECT DISTINCT S.sname
FROM   Sailors S
WHERE S.age>18 OR
    S.rating<=8;
```

- Get unique sailor names and show their age next year.

```
SELECT DISTINCT S.sname AS Name, S.age + 1 AS age_next_year
FROM   Sailors S;
```

Notes: Alias for columns

SELECT [DISTINCT] <columns list| *>
FROM <relations list>
[WHERE <conditions| AND| OR>] ;

Simple Queries (5) - Mathematical

- Get sailor names whose age is twice their rating

```
SELECT S.sname
FROM   Sailors S
WHERE S.age= 2* S.rating;
```

- Get sailor names whose twice their age is 3 times their rating

```
SELECT S.sname
FROM   Sailors S
WHERE 2*S.age= 3* S.rating;
```

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>] ;
```

Simple Queries (6) - String

- Find ages of sailors whose name begin and end with B and at least 3 characters long.

```
SELECT S.age
FROM   Sailors S
WHERE S.sname LIKE 'B_%B';
```

- Find ages of sailors whose name contain the letter a

```
SELECT S.sname
FROM   Sailors S
WHERE S.sname LIKE '%a%';
```

- Find ages of sailors whose name is 3 characters long

```
SELECT S.sname
FROM   Sailors S
WHERE S.sname LIKE '_ _ _';
```

Simple Queries (7) - Sorting

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Get all data from Sailors names sorted alphabetically

```
SELECT *
FROM   Sailors S
ORDER BY S.sname;
```

OR

```
SELECT *
FROM   Sailors S
ORDER BY S.sname ASC;
```

- Get Sailor names sorted descending

```
SELECT S.sname
FROM   Sailors S
ORDER BY S.sname DESC;
```

OR

```
SELECT S.sname
FROM   Sailors S
ORDER BY 1 DESC;
```

- Get Sailors sorted by name then by rating

```
SELECT S.sname
FROM   Sailors S
ORDER BY S.sname, S.rating ;
```

Join Queries

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find names of sailors who reserved boat id = 103.

```
SELECT S.sname
FROM   Sailors S, Reserves R
WHERE S.sid = R.sid AND
    R.bid = 103;
```

- OR

```
SELECT S.sname
FROM   Sailors S JOIN Reserves R ON S.sid = R.sid
WHERE R.bid = 103;
```

Join Queries (2)

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find sids of sailors who reserved a red boat.

```
SELECT Distinct R.sid
FROM Reserves R, Boats B
WHERE B.bid = R.bid AND
    B.color = ‘red’;
```

- Find names of sailors over 18 who reserved a red boat.

```
SELECT DISTINCT S.sname
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND
    B.bid = R.bid AND
    S.age>18 AND
    B.color = ‘red’;
```

Join Queries (3)

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find names of sailors who reserved at least one boat

```
SELECT DISTINCT S.sname
FROM Reserves R, Sailors S
WHERE S.sid = R.sid;
```

- Find colors of boats reserved by Lubber.

```
SELECT DISTINCT B.color
FROM Sailors S, Reserves R, Boats B
WHERE  S.sid = R.sid AND
    B.bid = R.bid AND
    S.sname = 'Lubber';
```

Nested Queries

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find names of sailors who have reserved a red boat

```
SELECT S.sname
FROM Sailors S
WHERE S.sid IN
   (SELECT R.sid
   FROM Reserves R
   WHERE R.bid IN
         (SELECT B.bid
         FROM Boats B
         WHERE B.color = 'red');
```

Nested Queries (2)

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find names of sailors who have not reserved a red boat

```
SELECT DISTINCT S.sname
FROM Sailors S
WHERE S.sid NOT IN
    (SELECT R.sid
    FROM Reserves R
    WHERE R.bid IN
            (SELECT B.bid
            FROM Boats B
            WHERE B.color = 'red');
```

Nested Queries (3) - Correlated

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find names of sailors who have reserved a boat 103

```
SELECT DISTINCT S.sname
FROM Sailors S
WHERE EXISTS
   (SELECT *
   FROM Reserves R
   WHERE R.bid = 103 AND
         R.sid = S.sid);
```

Set Operations Queries

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find names of sailors who reserved a red or green boat

```
SELECT DISTINCT S.sname
FROM Reserves R, Sailors S, Boats B
WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    B.color='red'
UNION
SELECT DISTINCT S.sname
FROM Reserves R, Sailors S, Boats B
WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    B.color='green';
```

```
SELECT DISTINCT S.sname
FROM Reserves R, Sailors S, Boats B
WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    (B.color='red' OR B.color='green');
```

Set Operations Queries (2)

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find names of sailors who reserved a red AND green boat

```
SELECT DISTINCT S.sname
FROM Reserves R, Sailors S, Boats B
WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    B.color='red'
INTERSECT
SELECT DISTINCT S.sname
FROM Reserves R, Sailors S, Boats B
WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    B.color='green';
```

```
SELECT DISTINCT S.sname
FROM Reserves R, Sailors S, Boats B
WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    B.color='red' AND
    S.sid IN
    (SELECT S.sid
    FROM Reserves R, Sailors S, Boats B
    WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    B.color='green' );
```

Set Operations Queries (3)

```
SELECT [DISTINCT] <columns list| *>
FROM   [relations list]
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find names of sailors who reserved a red but not green boat

```
SELECT DISTINCT S.sname
FROM Reserves R, Sailors S, Boats B
WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    B.color='red'
EXCEPT
SELECT DISTINCT S.sname
FROM Reserves R, Sailors S, Boats B
WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    B.color='green';
```

```
SELECT DISTINCT S.sname
FROM Reserves R, Sailors S, Boats B
WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    B.color='red' AND
    S.sid NOT IN
    (SELECT S.sid
    FROM Reserves R, Sailors S, Boats B
    WHERE S.sid = R.sid AND
    R.bid = B.bid AND
    B.color='green' );
```

Set Comparison Queries

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find names of sailors whose rating is better than some sailor named ‘Randy’

```
SELECT DISTINCT S.sname
FROM Sailors S
WHERE S.rating> ANY
    (SELECT S2.rating
    FROM Sailors S2
    WHERE S2.sname = ‘Randy’);
```

Combining ideas

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];
```

- Find names of sailors reserved all boats

```
SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS
   (SELECT B.bid
   FROM Boats B AND
   B.bid NOT IN
   (SELECT R.bid
   FROM Reserves R
   WHERE R.sid = S.sid);
```

SELECT [DISTINCT] <columns list| *>
FROM <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];

Aggregate operators

- How many Sailors do we have?

```
SELECT COUNT(*)
FROM Sailors S;
```

- How many Sailors over 18 do we have?

```
SELECT COUNT(*)
FROM Sailors S
WHERE S.age>18;
```

- How many different Sailor ratings do we have?

```
SELECT COUNT(DISTINCT S.rating)
FROM Sailors S;
```

- What is the average age of Sailors?

```
SELECT AVG(S.age)
FROM Sailors S;
```

SELECT [DISTINCT] <columns list| *>
FROM <relations list>
[WHERE <conditions| AND| OR>]
[ORDER BY <columns> [ASC|DESC]];

Aggregate operators (2)

- What is the maximum age of Sailors?

```
SELECT MAX(S.age)
FROM Sailors S;
```

- What is the minimum age of Sailors?

```
SELECT MIN(S.age)
FROM Sailors S;
```

- What is the sum of ages of Sailors?

```
SELECT SUM(S.age)
FROM Sailors S;
```

- Find the name of the oldest Sailor?

```
SELECT S.sname
FROM Sailors S
WHERE S.age =
(SELECT MAX(S.age)
FROM Sailors S);
```

```
SELECT S.sname, MAX(S.age)
FROM Sailors S;
```

Group By

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[GROUP BY <columns>
[ORDER BY <columns> [ASC|DESC]];
```

- What is the maximum age of each sailor rating?

```
SELECT MAX(S.age), S.rating
FROM Sailors S
GROUP BY S.rating;
```

- What is the maximum age of each sailor rating sorted by rating?

```
SELECT MAX(S.age), S.rating
FROM Sailors S
GROUP BY S.rating
ORDER BY 2;
```

- For each red boat, find the number of reservations?

```
SELECT B.bid, COUNT(*) AS reservationcount
FROM Boats B, Reserves R
WHERE  R.bid = B.bid AND
    B.color = 'red'
GROUP BY B.bid;
```

Having

```
SELECT [DISTINCT] <columns list| *>
FROM   <relations list>
[WHERE <conditions| AND| OR>]
[GROUP BY <columns>
   [HAVING <conditions>] ]
[ORDER BY <columns> [ASC|DESC]];
```

- Find the average age of sailors in each rating group with at least two sailors. Sort by rating descending.

```
SELECT AVG(S.age), S.rating
FROM Sailors S
GROUP BY S.rating
HAVING COUNT(*)>1
ORDER BY 2 DESC;
```

Find the average age of sailors who are of voting age (i.e., at least 18years old) for each rating level that has at least two sailors.

- Find the average age of sailors who are of voting age (i.e., at least 18years old) for each rating level that has at least two sailors.
- SELECT S.rating, AVG (S.age) AS avgage

FROM Sailors S

WHERE S. age >= 18

GROUP BY S.rating

HAVING 1 < (SELECT COUNT (*)

FROM Sailors S2

WHERE S.rating = S2.rating)