Introduction to Databases

SQL Statements
Chap. 5.1 – 5.5

SQL
• Structured Query Language (SQL)
  – Widely used
  – SQL:1999 is the latest standard

  – DDL: Data Definition Language
  – DML: Data Manipulation Language
  – Triggers
  – Embedded and Dynamic SQL
  – Client-server execution and remote access
  – Transaction management
  – Security
Structured Query Language (SQL)

- DDL: Data Definition Language
- DML: Data Manipulation Language
- Triggers
- Embedded and Dynamic SQL
- Client-server execution and remote access
- Transaction management
- Security

Select

```
SELECT [DISTINCT] columns FROM table_name [WHERE condition] [ORDER BY column(s)]
```

- Select
- Insert
- Update
- Delete

```
Select

• Basic form
  – SELECT [DISTINCT] select-list
  – FROM from-list
  – WHERE qualification

SELECT

• Sailors(sid:integer, sname:string,rating:integer,age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find the names and ages of all sailors
  – SELECT DISTINCT S.sname, S.age
  – FROM Sailors S
SELECT

- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

- Find the names and ages of all sailors
  - SELECT S.sname, S.age
  - FROM Sailors S

SELECT

- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

- Find all sailors with a rating over 7
  - SELECT S.sid, S.sname, S.rating, S.age
  - FROM Sailors AS S
  - WHERE S.rating > 7
SELECT

• Sailors(sid:integer, sname:string,rating:integer,age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find all sailors with a rating over 7
  – SELECT *
  – FROM Sailors AS S
  – WHERE S.rating > 7

SELECT

• Sailors(sid:integer, sname:string,rating:integer,age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find all sailors with a rating over 7
  – SELECT *
  – FROM Sailors
  – WHERE rating > 7

SELECT

• What really happens?
  1. Compute the cross-product of the tables in the from-list
  2. Delete rows that fail the qualification conditions
  3. Delete all columns that do not appear in the select-list
  4. If DISTINCT is specified, eliminate duplicate rows
SELECT

- Sailors(sid: integer, sname: string, rating: integer, age: real)
- Boats(bid: integer, bname: string, color: string)
- Reserves(sid: integer, bid: integer, day: date)

Find the names of the sailors who have reserved boat 103

- SELECT S.sname
  - FROM Sailors S, Reserves R
  - WHERE S.sid=R.sid AND R.bid = 103
SELECT
- Sailors(sid:integer, sname:string,rating:integer,age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

Find the names of the sailors who have reserved boat 103
- SELECT S.sname
- FROM Sailors S, Reserves R
- WHERE R.bid = 103

SELECT
- Sailors(sid:integer, sname:string,rating:integer,age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

Find the names of the sailors who have reserved boat 103
- SELECT sname
- FROM Sailors, Reserves
- WHERE Sailors.sid=Reserves.sid AND bid = 103

SELECT
- Sailors(sid:integer, sname:string,rating:integer,age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

Find the ID’s of the sailors who have reserved a red boat
SELECT

- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

- Find the ID’s of the sailors who have reserved a red boat
  - SELECT R.sid
  - FROM Boats B, Reserves R
  - WHERE Boats.bid=Reserves.bid AND B.color='red'

SELECT

- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

- Find the names of the sailors who have reserved a red boat
  - SELECT S.Sname
  - FROM Sailors S, Boats B, Reserves R
  - WHERE S.sid=R.sid AND B.bid=R.bid AND
  - B.color='red'
SELECT

- Sailors(sid: integer, sname: string, rating: integer, age: real)
- Boats(bid: integer, bname: string, color: string)
- Reserves(sid: integer, bid: integer, day: date)

Find the names of the sailors who have reserved at least one boat
- SELECT S.Sname
- FROM Sailors S, Reserves R
- WHERE S.sid=R.sid

SELECT

- Sailors(sid: integer, sname: string, rating: integer, age: real)
- Boats(bid: integer, bname: string, color: string)
- Reserves(sid: integer, bid: integer, day: date)

Find the names of the sailors who have reserved at least one boat
- SELECT DISTINCT S.Sname
- FROM Sailors S, Reserves R
- WHERE S.sid=R.sid

SELECT

- Sailors(sid: integer, sname: string, rating: integer, age: real)
- Boats(bid: integer, bname: string, color: string)
- Reserves(sid: integer, bid: integer, day: date)

Find the names of the sailors who have reserved at least one boat
- SELECT DISTINCT S.Sname
- FROM Sailors S, Reserves R
- WHERE S.sid=R.sid
SELECT
• Sailors(sid:integer, sname:string, rating:integer, age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Compute increments for the rating of person who have saided two different boats on the same day

SELECT
• Sailors(sid:integer, sname:string, rating:integer, age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Compute increments for the rating of person who have saided two different boats on the same day
  – SELECT S.sname, S.rating+1 AS rating
  – FROM Sailors S, Reserves R1, Reserves R2
  – WHERE S.sid = R1.sid AND S.sid = R2.sid AND
    R1.day = R2.day AND R1.bid <> R2.bid

SELECT
• Sailors(sid:integer, sname:string, rating:integer, age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find the ages of sailors whose name begins and ends with B and has at least three characters
• Find the ages of sailors whose name begins and ends with B and has at least three characters
  – SELECT S.age
  – FROM Sailors S
  – WHERE S.sname LIKE 'B_%B'
SELECT

• Sailors(sid:integer, sname:string, rating:integer, age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find the names of sailors whose have reserved a red or a green boat

– SELECT S.sname
– FROM Sailors S, Reserves R, Boats B
  (B.color='red' or B.color='green')

SELECT

• Sailors(sid:integer, sname:string, rating:integer, age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find the names of sailors whose have reserved both a red and a green boat

• Find the names of sailors whose have reserved both a red and a green boat
SELECT

• Sailors(sid:integer, sname:string, rating:integer, age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find the names of sailors whose have reserved both a red and a green boat
  - SELECT S.sname
  - FROM Sailors S, Reserves R1, Reserves R2, Boats B1, Boats B2

SELECT

• Sailors(sid:integer, sname:string, rating:integer, age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find the names of sailors whose have reserved both a red and a green boat
  - SELECT S1.sname
  - FROM Sailors S1, Reserves R1, Boats B1
  - WHERE S1.sid = R1.sid AND R1.bid = B1.bid AND B1.color='red'
  - INTERSECT
  - SELECT S2.sname
  - FROM Sailors S2, Reserves R2, Boats B2
  - WHERE S2.sid = R2.sid AND R2.bid = B2.bid AND B2.color='green'

SELECT

• Sailors(sid:integer, sname:string, rating:integer, age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find the names of sailors whose have reserved both a red or a green boat
  - SELECT S1.sname
  - FROM Sailors S1, Reserves R1, Boats B1
  - WHERE S1.sid = R1.sid AND R1.bid = B1.bid AND B1.color='red'
  - UNION
  - SELECT S2.sname
  - FROM Sailors S2, Reserves R2, Boats B2
  - WHERE S2.sid = R2.sid AND R2.bid = B2.bid AND B2.color='green'
SELECT

• Sailors(sid:integer, sname:string, rating:integer, age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find the sids of all sailors whose have reserved a red boat but not a green boat
  – SELECT R1.sid
  – FROM Reserves R1, Boats B1
  – WHERE R1.bid = B1.bid AND B1.color='red'
  – EXCEPT
  – SELECT R2.sid
  – FROM Reserves R2, Boats B2

SELECT: Nested Queries

• Sailors(sid:integer, sname:string, rating:integer, age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find the sids of all sailors who have reserved boat 104
  – SELECT S.sid
  – FROM Sailors S
  – WHERE S.rating = 10
  – UNION
  – SELECT R.sid
  – FROM Reserves R
  – WHERE R.bid =104

• Find the sids of all sailors who have reserved boat 103
  – SELECT S.sname
  – FROM Sailors S, Reserves R
  – WHERE S.sid = R.sid AND R.bid = 103
SELECT: Nested Queries

- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

- Find the sids of all sailors who have reserved boat 103
  - SELECT S.sid
  - FROM Sailors S
  - WHERE S.sid IN (SELECT R.sid
    FROM Reserves R
    WHERE R.bid = 103)

SELECT: Nested Queries

- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

- Find the sids of all sailors who have not reserved boat 103
  - SELECT S.sid
  - FROM Sailors S
  - WHERE S.sid NOT IN (SELECT R.sid
    FROM Reserves R
    WHERE R.bid = 103)

SELECT: Nested Queries

- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

- Find the names of sailors who have reserved a red boat
  - SELECT S.sname
  - FROM Sailors S
  - WHERE S.sname IN (SELECT B.bname
    FROM Boats B
    WHERE B.color = 'red')
SELECT: Nested Queries

- Sailors(sid:integer, sname:string,rating:integer,age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

• Find the 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 BOAT B
  WHERE B.color = 'red') )

• Find the names of sailors who have not reserved a red boat
  - SELECT S.sname
  - FROM Sailors S
  - WHERE S.sid NOT IN (SELECT R.sid
  FROM Reserves R
  WHERE R.bid IN (SELECT B.bid
  FROM BOAT B
  WHERE B.color = 'red') )
SELECT: Nested Queries

- Sailors(sid:integer, sname:string,rating:integer,age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

• Find the names of sailors who have reserved boat 103
  – SELECT S.sname
  – FROM Sailors S
  – WHERE EXISTS (SELECT *
          FROM Reserves R
          WHERE R.bid = 103 AND R.sid = S.sid )

• Find the names of sailors who have not reserved boat 103
  – SELECT S.sname
  – FROM Sailors S
  – WHERE NOT EXISTS (SELECT *
          FROM Reserves R
          WHERE R_bid = 103 AND R.sid = S.sid )

• Find the sailors whose rating is better than Horatio
  – SELECT S.sname
  – FROM Sailors S
  – WHERE rating > (SELECT DISTINCT rating
          FROM Sailors
          WHERE sname = 'Horatio')
SELECT: Nested Queries

- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

- Find the sailors whose rating is better than Horatio
  - SELECT S.sid
  - FROM Sailors S
  - WHERE S.rating > ANY (SELECT S2.rating
    FROM Sailors S2
    WHERE S2.sname = 'Horatio' )

- Find the sailors with the highest rating
  - SELECT S.sid
  - FROM Sailors S
  - WHERE S.rating >= ALL (SELECT S2.rating
    FROM Sailors S2)
SELECT: Nested Queries

- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

Find the names of sailors who have reserved all boats

```
SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS (SELECT B.Bid
                   FROM Boats B
                   EXCEPT
                   (SELECT R.bid
                    FROM Reserves R
                    WHERE R.sid = S.sid)
                   )
```

Find the names of sailors who have reserved all boats

```
SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS (SELECT B.Bid
                   FROM Boats B
                   WHERE NOT EXISTS (SELECT R.bid
                                      FROM Reserves R
                                      WHERE R.sid = S.sid)
                   )
```
### SELECT: AGGREGATE OPERATORS

- `COUNT([DISTINCT] A)`
- `SUM([DISTINCT] A)`
- `AVG([DISTINCT] A)`
- `MAX([DISTINCT] A)`
- `MIN([DISTINCT] A)`

### SELECT: Aggregate Operators

- Sailors(`sid`:integer, `sname`:string,`rating`:integer,`age`:real)
- Boats(`bid`:integer, `bname`: string, `color`: string)
- Reserves(`sid`:integer, `bid`: integer, `day`: date)

Find the average age of sailors with a rating of 10:

```sql
SELECT AVG(S.age)
FROM Sailors S
WHERE S.rating = 10
```
SELECT:
Aggregate Operators
- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

- Find the name and age of the oldest sailor
  - SELECT S.sname, MAX(S.age)
  - FROM Sailors S

- Find the name and age of the oldest sailor
  SELECT S.sname
  FROM Sailors S
  WHERE S.age = (SELECT MAX(S2.age)
                  FROM Sailors S2)
SELECT:
Aggregate Operators
• Sailors(sid:integer, sname:string,rating:integer,age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Count the number of sailors

SELECT COUNT (*)
FROM Sailors S

SELECT:
Aggregate Operators
• Sailors(sid:integer, sname:string,rating:integer,age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Count the number of different sailor names
SELECT:
Aggregate Operators

• Sailors(sid:integer, sname:string,rating:integer,age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Count the number of different sailor names
  SELECT COUNT (DISTINCT S.sname)
  FROM Sailors S

• Find the names of sailors who are older than the oldest sailor with a rating of 10
  SELECT S.sname
  FROM Sailors S
  WHERE S.age > (SELECT MAX (S2.age)
  FROM Sailors S2
  WHERE S2.rating = 10)
• Sailors(sid:integer, sname:string,rating:integer,age:real)
• Boats(bid:integer, bname: string, color: string)
• Reserves(sid:integer, bid: integer, day: date)

• Find the age of the youngest sailor for each rating level
  SELECT S.rating, MIN (S.age)
  FROM Sailors S
  GROUP BY S.rating

• Find the age of the youngest sailor
  SELECT MIN (S.age)
  FROM Sailors S

• Find the age of the youngest sailor who is eligible to vote, i.e., older than 18) for each level of rating with at least two such sailors
SELECT: GROUP BY and HAVING Clauses

- Sailors(sid:integer, sname: string, rating: integer, age: real)
- Boats(bid: integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

Find the age of the youngest sailor who is eligible to vote, i.e., older than 18) for each level of rating with at least two such sailors

```sql
SELECT S.rating, MIN(S.age) AS minage
FROM Sailors S
WHERE S.age >= 19
GROUP BY S.rating
HAVING COUNT(*) > 1
```

SELECT: GROUP BY and HAVING Clauses

- Sailors(sid:integer, sname: string, rating: integer, age: real)
- Boats(bid: integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

Find the age of the youngest sailor who is eligible to vote, i.e., older than 18) for each level of rating with at least two such sailors

```sql
SELECT S.rating, MIN(S.age) AS minage
FROM Sailors S
WHERE S.age >= 19
GROUP BY S.rating
HAVING COUNT(*) > 1
```

Step 1

```sql
SELECT S.rating, MIN(S.age) AS minage
FROM Sailors S
WHERE S.age >= 19
GROUP BY S.rating
HAVING COUNT(*) > 1
```

Step 2

```sql
SELECT S.rating, MIN(S.age) AS minage
FROM Sailors S
WHERE S.age >= 19
GROUP BY S.rating
HAVING COUNT(*) > 1
```
SELECT: GROUP BY and HAVING Clauses

- Sailors(sid:integer, sname:string, rating:integer, age:real)
- Boats(bid:integer, bname: string, color: string)
- Reserves(sid:integer, bid: integer, day: date)

Find the age of the youngest sailor who is eligible to vote, i.e., older than 18) for each level of rating with at least two such sailors

```
SELECT S.rating, MIN (S.age) AS minage
FROM Sailors S
WHERE S.age >= 19
GROUP BY S.rating
HAVING COUNT(*) > 1
```

Step 3

Step 4

Find the age of the youngest sailor who is eligible to vote, i.e., older than 18) for each level of rating with at least two such sailors

```
SELECT S.rating, MIN (S.age) AS minage
FROM Sailors S
WHERE S.age >= 19
GROUP BY S.rating
HAVING COUNT(*) > 1
```

Step 5
Step 6

Find the age of the youngest sailor who is eligible to vote, i.e., older than 18) for each level of rating with at least two such sailors

```
SELECT S.rating, MIN(S.age) AS minage
FROM Sailors S
WHERE S.age >= 19
GROUP BY S.rating
HAVING COUNT(*) > 1
```

For each boat, find the number of reservations for the red boat

```
SELECT B.bid, COUNT(*) AS reservationcount
FROM Boats B, Reserve R
WHERE B.bid = R.bid AND B.color = 'R'
GROUP BY B.bid
```
### SELECT: GROUP BY and HAVING Clauses

- **Sailors** *(sid: integer, sname: string, rating: integer, age: real)*
- **Boats** *(bid: integer, bname: string, color: string)*
- **Reserves** *(sid: integer, bid: integer, day: date)*

Find the average age of sailors for each rating level that has at least two sailors.

#### Query:

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

Find those ratings for which the average age of the sailors is the minimum over all ratings.
• Sailors(sid: integer, sname: string, rating: integer, age: real)
• Boats(bid: integer, bname: string, color: string)
• Reserves(sid: integer, bid: integer, day: date)

Find those ratings for which the average age of the sailors is the minimum over all ratings

```sql
SELECT S.rating
FROM Sailors S
WHERE AVG(S.age) = (SELECT MIN(AVG(S2.age))
FROM Sailors S2
GROUP BY S2.rating)
```
Any Questions?