

# BOB36DBS: Database Systems | Classes 5 and 6: SQL: Data Querying

---

## 01: Department Teachers

```
SELECT *
FROM teacher
WHERE (department = "KSI");
```

```
SELECT id, name, phone, department ...

... WHERE department = "KSI"
```

---

## 02: Study Results

```
SELECT code, title, result
FROM Enrollment NATURAL JOIN Course
WHERE (student = 4301) AND (semester = 201)
ORDER BY result, title DESC;
```

```
... FROM Enrollment NATURAL INNER JOIN Course ...
```

```
... FROM Enrollment AS E JOIN Course AS C ON (E.code = C.code) ...
```

```
... FROM Enrollment AS E JOIN Course AS C
WHERE (E.code = C.code) AND (student = 4301) AND (semester = 201) ...
```

```
... WHERE (E.code = C.code) AND (E.student = 4301) AND (E.semester = 201) ...
```

```
... FROM Enrollment AS E, Course AS C
WHERE (E.code = C.code) AND (student = 4301) AND (semester = 201) ...
```

```
... ORDER BY result ASC, name DESC
```

---

## 03: Subordinate Teachers

```
SELECT T2.name
FROM
  Teacher AS T1
  JOIN Department AS D ON (T1.id = D.chair)
  JOIN Teacher AS T2 ON (D.name = T2.department)
WHERE (T1.name = "Tomáš Skopal");
```

```
SELECT T2.name
FROM
  Teacher AS T2
  JOIN Department AS D ON (T2.department = D.name)
  JOIN Teacher AS T1 ON (D.chair = T1.id)
WHERE (T1.name = "Tomáš Skopal");
```

```
SELECT T2.name
FROM Teacher AS T1 JOIN Department AS D JOIN Teacher AS T2
WHERE (T1.name = "Tomáš Skopal") AND (T1.id = D.chair) AND (D.name = T2.department);
```

---

## 04: Permitted Courses

```
SELECT DISTINCT code, title
FROM
  Schedule JOIN
  Course ON (course = code)
WHERE (semester = 202) AND (day = "MON" OR day = "FRI");
```

```
SELECT DISTINCT C.code, C.title
FROM
  Schedule AS S
  JOIN Course AS C ON (S.course = C.code)
WHERE (S.semester = 202) AND (S.day = "MON" OR S.day = "FRI");
```

```
... WHERE (semester = 202) AND (day IN ("MON", "FRI"))
```

```
SELECT code, title
FROM Course
WHERE
  code IN (
    SELECT DISTINCT course
    FROM Schedule
    WHERE (semester = 202) AND (day = "MON" OR day = "FRI")
  );
```

```
SELECT code, title
FROM Course
WHERE
  EXISTS (
    SELECT *
    FROM Schedule
    WHERE (course = code) AND (semester = 202) AND (day = "MON" OR day = "FRI")
  );
```

```
SELECT code, title
FROM Course
WHERE
  code = ANY (
    SELECT course
    FROM Schedule
    WHERE (semester = 202) AND (day = "MON" OR day = "FRI")
  );
```

```
... code = SOME (...) ...
```

```
SELECT code, title
FROM
  Course
  NATURAL JOIN (
    SELECT DISTINCT course AS code
    FROM Schedule
    WHERE (semester = 202) AND (day = "MON" OR day = "FRI")
  );
```

```

SELECT code, title
FROM Schedule JOIN Course ON (course = code)
WHERE (semester = 202) AND (day = "MON")
UNION
SELECT code, title
FROM Schedule JOIN Course ON (course = code)
WHERE (semester = 202) AND (day = "FRI");

```

... UNION DISTINCT ...

## 05: Prohibited Courses

Incorrect:

```

SELECT code, title
FROM Schedule JOIN Course ON (course = code)
WHERE (semester = 202) AND NOT (day = "MON" OR day = "FRI");

```

... WHERE (semester = 202) AND (day <> "MON" AND day <> "FRI")

... WHERE NOT((semester = 202) AND (day = "MON" OR day = "FRI"))

Correct:

```

SELECT code, title
FROM Course
WHERE
  code NOT IN (
    SELECT DISTINCT course
    FROM Schedule
    WHERE (semester = 202) AND (day = "MON" OR day = "FRI")
  );

```

```

SELECT code, title
FROM Course
WHERE
  NOT EXISTS (
    SELECT *
    FROM Schedule
    WHERE (course = code) AND (semester = 202) AND (day = "MON" OR day = "FRI")
  );

```

```

SELECT code, title
FROM Course
WHERE
  code <> ALL (
    SELECT course
    FROM Schedule
    WHERE (semester = 202) AND (day = "MON" OR day = "FRI")
  );

```

```

SELECT code, title
FROM Course
EXCEPT
SELECT code, title
FROM Schedule JOIN Course ON (course = code)
WHERE (semester = 202) AND (day = "MON" OR day = "FRI")

```

```

SELECT code, title
FROM Course
WHERE
  code IN (
    SELECT code
    FROM Course
  EXCEPT
    SELECT course AS code
    FROM Schedule
    WHERE (semester = 202) AND (day = "MON" OR day = "FRI")
  );

```

```

SELECT code, title
FROM
  Course
  LEFT OUTER JOIN Schedule ON
    (code = course) AND (semester = 202) AND (day = "MON" OR day = "FRI")
WHERE (course IS NULL);

```

Incorrect:

```

... WHERE (course = NULL)

```

## 06: Inactive Students

```

SELECT S.name, S.address
FROM Student AS S
WHERE
  NOT EXISTS (
    SELECT *
    FROM Enrollment AS E
    WHERE (E.student = S.id) AND (E.semester IN (201, 202))
  );

```

...

## 07: Promising Students

```

SELECT DISTINCT S.id, S.name
FROM
  Student AS S
  JOIN Enrollment AS E ON (S.id = E.student)
  JOIN Schedule AS U ON (E.code = U.course) AND (E.semester = U.semester)
  JOIN Teacher AS T ON (U.teacher = T.id)
WHERE (E.semester = 202) AND (T.department = "KSI");

```

```

SELECT DISTINCT S.id, S.name
FROM
  Student AS S JOIN Enrollment AS E JOIN Schedule AS U JOIN Teacher AS T
WHERE
  (E.semester = 202) AND (T.department = "KSI")
AND
  (S.id = E.student)
AND
  (E.code = U.course) AND (E.semester = U.semester)
AND
  (U.teacher = T.id);

```

```

SELECT S.id, S.name
FROM Student AS S
WHERE
  EXISTS (
    SELECT *
    FROM
      Enrollment AS E
      JOIN Schedule AS U ON (E.code = U.course) AND (E.semester = U.semester)
      JOIN Teacher AS T ON (U.teacher = T.id)
    WHERE (E.student = S.id) AND (E.semester = 202) AND (T.department = "KSI")
  );

```

```

SELECT S.id, S.name
FROM Student AS S
WHERE
  EXISTS (
    SELECT E.*
    FROM Enrollment AS E
    WHERE
      (E.student = S.id) AND (E.semester = 202) AND
      EXISTS (
        SELECT U.*
        FROM Schedule AS U
        WHERE
          (U.course = E.code) AND (U.semester = E.semester) AND
          EXISTS (
            SELECT T.*
            FROM Teacher AS T
            WHERE (T.id = U.teacher) AND (T.department = "KSI")
          )
        )
      )
  );

```

---

## 08: Loyal Students

```

SELECT S.id, S.name
FROM Student AS S
WHERE
  EXISTS (
    SELECT *
    FROM Enrollment AS E
    WHERE (E.student = S.id) AND (E.semester = 202)
  )
AND
  NOT EXISTS (
    SELECT *
    FROM
      Enrollment AS E
      JOIN Schedule AS U ON (E.code = U.course) AND (E.semester = U.semester)
      JOIN Teacher AS T ON (U.teacher = T.id)
    WHERE (E.student = S.id) AND (E.semester = 202) AND (T.department <> "KSI")
  );

```

```

SELECT DISTINCT id, name
FROM Student JOIN Enrollment ON (id = student)
WHERE
    (semester = 202) AND
    NOT EXISTS (
        SELECT *
        FROM
            Enrollment AS E
            JOIN Schedule AS U ON (E.code = U.course) AND (E.semester = U.semester)
            JOIN Teacher AS T ON (U.teacher = T.id)
        WHERE (E.student = S.id) AND (E.semester = 202) AND (T.department <> "KSI")
    );

```

---

## 09: Timetable Conflicts

```

SELECT T.name
FROM Teacher AS T
WHERE
    EXISTS (
        SELECT *
        FROM
            (Schedule AS U1 JOIN Room AS R1 ON (U1.room = R1.number))
            JOIN
            (Schedule AS U2 JOIN Room AS R2 ON (U2.room = R2.number))
            ON
                (U1.semester = U2.semester) AND
                (U1.teacher = U2.teacher) AND
                (U1.day = U2.day)
            AND
                (U1.time < U2.time)
        WHERE
            (U1.teacher = T.id) AND (U1.semester = 211)
            AND
            (
                ((R1.building = R2.building) AND (U1.time + 90 + 15 > U2.time))
                OR
                ((R1.building <> R2.building) AND (U1.time + 90 + 60 > U2.time))
            )
    );

```

```

SELECT DISTINCT T.name
FROM
    (Schedule AS U1 JOIN Room AS R1 ON (U1.room = R1.number))
    JOIN
    (Schedule AS U2 JOIN Room AS R2 ON (U2.room = R2.number))
    ON
        (U1.semester = U2.semester) AND
        (U1.teacher = U2.teacher) AND
        (U1.day = U2.day)
    AND
        (U1.time < U2.time)
    JOIN Teacher AS T ON (U1.teacher = T.id)
WHERE
    (U1.semester = 211)
    AND
    (
        ((R1.building = R2.building) AND (U1.time + 90 + 15 > U2.time))
        OR
        ((R1.building <> R2.building) AND (U1.time + 90 + 60 > U2.time))
    );

```

---

## 10-A: Room Statistics

```
SELECT AVG(capacity) AS average, COUNT(number) AS count
FROM Room;
```

```
SELECT AVG(ALL capacity) AS average, COUNT(ALL number) AS count ...
... COUNT(*) AS count ...
```

---

## 10-B: Building Statistics

```
SELECT building, SUM(capacity) AS sum
FROM Room
GROUP BY building;
```

---

## 11: Enrollment Statistics

```
SELECT title, COUNT(*) as students, AVG(result) AS average
FROM Course NATURAL JOIN Enrollment
WHERE (semester = 201)
GROUP BY code, title
HAVING (COUNT(*) >= 10)
ORDER BY average;
```