SQL tips and tricks

COMS10012 Software Tools

Rank queries

Top student(s)

```
SELECT name, grade FROM Student
WHERE grade =
    (SELECT MAX(grade) FROM Student);
```

not:

```
SELECT name, grade FROM Student
ORDER BY grade DESC LIMIT 1;
```

Ranks

What rank was the student with id = 21 within their cohort?

Student
*id name grade cohort

grade	rank
90	1
89	2
85	3
85	3
80	5

Ranks

1. Find all the students in the same cohort who got a strictly greater mark than student 21.
2. Count the number of entries in this table, and add 1.

Ranks

```
SELECT 1 + COUNT(1) FROM (
  SELECT id FROM Student
  WHERE grade >
    (SELECT grade FROM Student WHERE id = 21)
  AND cohort =
   (SELECT cohort FROM Student WHERE id = 21)
);
```

Ranks

```
SELECT 1 + COUNT(1) FROM (
  SELECT id FROM Student S
  INNER JOIN Student T
  ON S.cohort = T.cohort
  AND S.grade > T.grade
  WHERE T.id = 21
);
```

Set operations

Set operations

```
SELECT ...
    [UNION [ALL] | INTERSECT | EXCEPT ]
SELECT ...
```

requirement: all queries return the same number of columns. Column names don't matter.

UNION ALL

```
SELECT username, cohort FROM Student
WHERE cohort LIKE 'M%' AND grade >= 50

UNION ALL

SELECT username, cohort FROM Student
WHERE cohort NOT LIKE 'M%'
AND grade >= 40;
```

UNION ALL: "total" rows

```
SELECT cohort, AVG(grade) AS average
FROM Student GROUP BY cohort

UNION ALL

SELECT 'all', AVG(grade)
FROM Student;
```


UNION [ALL]

UNION ALL: stick these extra rows on the end.

This is what you want most of the time.

UNION: stick these rows on the end, then remove duplicates. This may cost a sort, e.g. O(n log n) where n is the total number of rows.

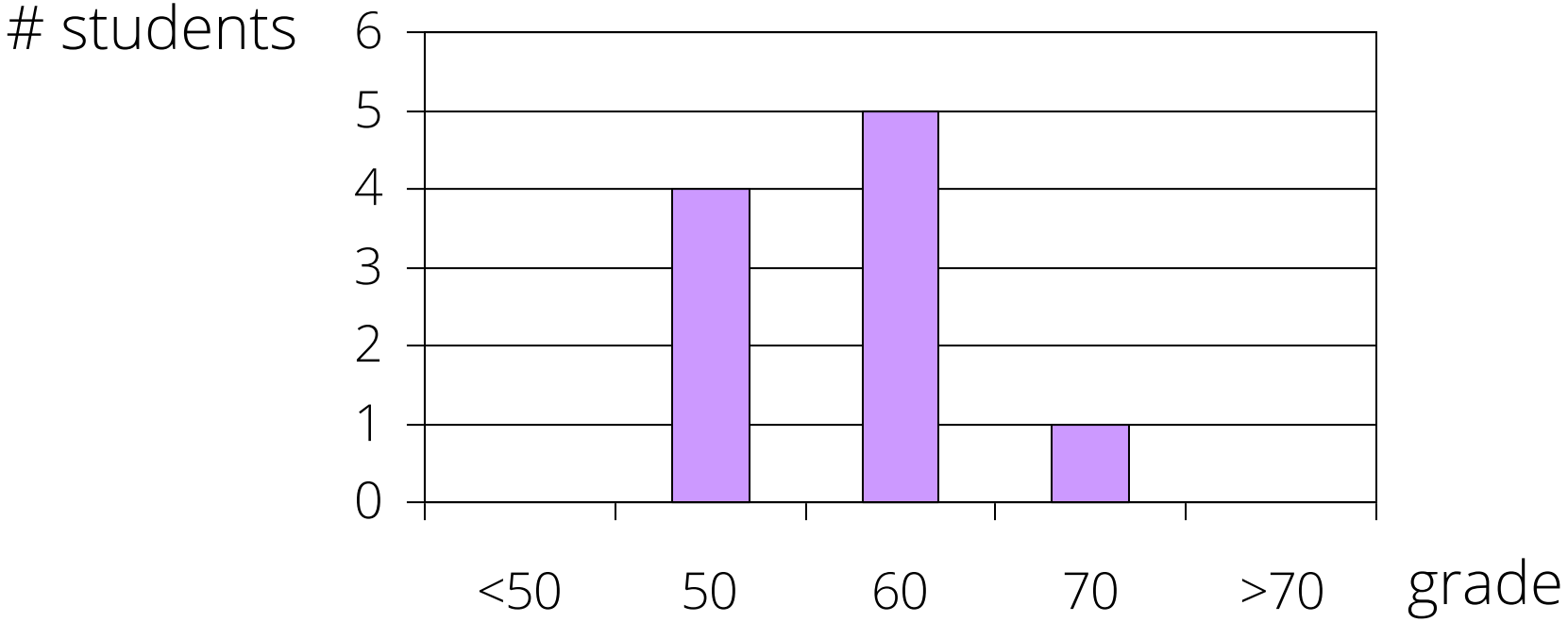
CASE

CASE

```
CASE <expression>
WHEN <value> THEN <value>
[WHEN <value> THEN <value> ...]
[ELSE <value>]
END
```

```
SELECT name,
CASE gender
   WHEN 1 THEN 'F'
   WHEN 0 THEN 'M'
   ELSE 'X' END
FROM Person;
```


Histogram



Histogram

```
SELECT grade, COUNT(1)
FROM Enrol
WHERE unit = ...
GROUP BY grade
ORDER BY grade;
```

Enrol
*student *unit grade

grade	count(1)
54	1
57	1
58	2
60	2
61	1
62	1
63	1
73	1

Histogram

```
SELECT (grade – mod(grade, 10)) AS g,
COUNT(1) AS c FROM Enrol
WHERE unit = ...
GROUP BY g ORDER BY g;
```

g	c
50	4
60	5
70	1

CASE statement

```
SELECT CASE
  WHEN grade >= 70 THEN 'First'
  WHEN grade >= 65 THEN '2:1'
  WHEN grade >= 60 THEN '2:2'
  ELSE 'Third' END
AS g ...
```


More CASE

```
SELECT unit, MIN(grade), MAX(grade),
AVG(grade), COUNT(1) AS N,
SUM(CASE WHEN grade >= 50
THEN 1 ELSE 0 END) AS pass
FROM Enrol GROUP BY unit;
```

unit	min	max	avg	N	pass
11	54	73	60.6	10	10
12	46	71	61.4	10	8
13	NULL	NULL	NULL	5	0
15	48	74	61.7	10	9