Ukážky syntaxe sql dotazov

Vytvorenie tabuliek CREATE

```
CREATE TABLE student
(
meno_krstne VARCHAR(20) NOT NULL,
meno_priezvisko VARCHAR(20) NOT NULL,
datum_narodenia DATE NOT NULL,
pohlavie ENUM('M', 'Z') NOT NULL,
student_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY
);

CREATE TABLE udalost
(
datum DATE NOT NULL,
typ ENUM('T', 'Z') NOT NULL,
udalost_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY
);

CREATE TABLE skore
(
student_id INT UNSIGNED NOT NULL,
udalost_id INT UNSIGNED NOT NULL,
PRIMARY KEY (udalost_id, student_id),
skore INT NOT NULL
);
```

Naplnenie tabuliek INSERT

```
INSERT INTO student VALUES ( 'Janko', 'Hraško', '1989-7-3', 'M', '');

INSERT INTO student (meno_krstne, meno_priezvisko, datum_narodenia,
pohlavie) VALUES ( 'Jurko', 'Hraško', '1989-8-4', 'M');

INSERT INTO student SET meno_krstne='Anička',
meno_priezvisko='Študentka', datum_narodenia='1988-12-6',
pohlavie='Z';
INSERT INTO student SET meno_krstne='Martina',
meno_priezvisko='Spolužiačka', datum_narodenia='1988-11-3',
pohlavie='Z';

INSERT INTO udalost SET datum='12-9-2008', typ='Z';
INSERT INTO udalost SET datum='2008-7-3', typ='T';
INSERT INTO udalost SET datum='2008-4-3', typ='Z';

INSERT INTO skore VALUES ('1','2','15');
INSERT INTO skore VALUES ('2','2','20');
INSERT INTO skore VALUES ( '3', '2', '25');
INSERT INTO skore VALUES ( '1', '3', '15');
INSERT INTO skore VALUES ( '2', '3', '15');
INSERT INTO skore VALUES ( '3', '3', '15');
INSERT INTO skore VALUES ('1','1','15');
```

Výber SELECT z tabuliek

```
SELECT #čo sa má vybrať, názvy stĺpcov
FROM #zoznam tabuliek
WHERE #primárne obmedzenie, aké podmienky musia záznamy spĺňať
GROUP BY #stĺpce podľa ktorých sa budú výsledky zoskupovať
ORDER BY #stĺpce podľa ktorých sa budú výsledky radiť
HAVING #sekundárne obmedzenia
LIMIT #obmedzenie počtu výsledkov

SELECT * FROM student;

SELECT * FROM student WHERE pohlavie='M';

SELECT meno_krstne FROM student;

SELECT * FROM student WHERE datum_narodenia>='1989-1-1';

SELECT * FROM student WHERE MONTH(datum_narodenia)='7';

SELECT * FROM student WHERE MONTH(datum_narodenia)=MONTH(CURDATE())
AND DAYOFMONTH(datum_narodenia)=DAYOFMONTH(CURDATE()) ;

SELECT * FROM student WHERE meno_priezvisko='Hraško';
SELECT * FROM student WHERE meno_priezvisko LIKE 'H%';

SELECT * FROM student ORDER BY datum_narodenia;

SELECT COUNT(*) FROM student;

SELECT * FROM udalost;

SELECT * FROM skore WHERE skore >= 20;

SELECT pohlavie, COUNT(*) AS 'počet' FROM student GROUP BY pohlavie
ORDER BY 'počet';


SELECT udalost_id,
Min(skore) AS 'Minimum',
Max(skore) AS 'Maximum',
SUM(skore) AS 'Súčet',
AVG(skore) AS 'Priemer',
COUNT(skore) AS 'Počet'
FROM skore
GROUP BY udalost_id;


SELECT student.meno_krstne, student.meno_priezvisko,
skore.skore, udalost.typ, udalost.datum
FROM student, udalost, skore;

#kartézsky súčin každý údaj každej tabuľky s každým údajom každej
tabuľky
#podmienka WHERE spojí tabuľky takým spôsobom, že spojí navzájom si
odpovedajúce riadky tabuliek, kde obe obsahujú rovnaký údaj.
```

WHERE môže použiť IF EXISTS a IF NOT EXISTS

```
SELECT student.meno_krstne, student.meno_priezvisko, skore.skore,
udalost.typ, udalost.datum
FROM student, udalost, skore
WHERE udalost.udalost_id= skore.udalost_id
AND skore.student_id= student.student_id;

SELECT skore.student_id, udalost.datum, skore.skore, udalost.typ
FROM udalost,skore
WHERE udalost.datum= '2008-7-3'
AND udalost.udalost_id=skore.udalost_id;

SELECT *FROM  udalost , student
LEFT JOIN skore
ON student.student_id=skore.student_id
WHERE udalost.udalost_id=skore.udalost_id
ORDER BY student.student_id, udalost.udalost_id;

SELECT *FROM udalost
LEFT JOIN skore
ON udalost.udalost_id=skore.udalost_id;
```

JOIN LEFT RIGHT INNER OUTER NATURAL

```
SELECT *FROM udalost
LEFT JOIN skore
ON udalost.udalost_id=skore.udalost_id;

SELECT *FROM udalost
LEFT JOIN skore
USING (udalost_id);

SELECT *FROM udalost
FULL JOIN skore
USING (udalost_id);

SELECT *FROM udalost
CROSS JOIN skore
ON udalost.udalost_id=skore.udalost_id;

SELECT *FROM udalost
INNER JOIN skore
ON udalost.udalost_id=skore.udalost_id;

SELECT *FROM udalost
LEFT JOIN skore
ON udalost.udalost_id=skore.udalost_id
ORDER BY skore.student_id;

SELECT *FROM student, udalost
LEFT JOIN skore
ON udalost.udalost_id=skore.udalost_id
WHERE skore.student_id= student.student_id
```

```
ORDER BY skore.student_id;

SELECT *FROM udalost, student
LEFT JOIN skore
ON student.student_id=skore.student_id
WHERE udalost.udalost_id=skore.udalost_id
ORDER BY skore.student_id;

SELECT *FROM student, skore
LEFT JOIN udalost
UNION udalost_id
WHERE skore.student_id=student.student_id
ORDER BY skore.student_id;

SELECT *FROM udalost
LEFT JOIN skore
ON udalost.udalost_id=skore.udalost_id
WHERE skore.skore IS NULL
ORDER BY skore.student_id;
```

Zložité SELEKTY poddotazy IN a NOT IN

```
SELECT * FROM skore
WHERE udalost_id IN (SELECT udalost_id FROM udalost WHERE typ='Z');

SELECT * FROM skore
WHERE udalost_id NOT IN (SELECT udalost_id FROM udalost WHERE
typ='Z');

SELECT MAX(skore)FROM skore WHERE udalost_id=(SELECT udalost_id FROM
udalost WHERE datum='2008-4-3')
AND udalost_id=

SELECT * FROM skore WHERE skore=(
(SELECT udalost_id FROM udalost WHERE datum='2008-4-3')
);

SELECT skore FROM skore, udalost
WHERE skore.udalost_id = udalost.udalost_id AND udalost.typ='Z';
```

UPDATE

```
UPDATE skore
SET
student_id ='3',
udalost_id ='2',
skore='30'
WHERE student_id ='3'
AND udalost_id ='2';
```

DELETE

```
DELETE  FROM student
WHERE meno_priezvisko='Spolužiačka'
AND meno_krstne='Martina';
```