

SQL Cheat sheet GCSE/ALEVEL/DEGREE

```
//basic syntax
SELECT //list the fields to be shown
//or * wildcard for all
FROM //list the table
WHERE //list the search criteria
ORDER BY //default is ascending order
```

```
//creating a table named student
CREATE TABLE student (
  id INTEGER UNIQUE PRIMARY KEY,
  name VARCHAR(20),
  grade INTEGER NOT NULL,
  age INTEGER,
  joinDate DATE,
  fee CURRENCY
);
```

```
//add a column(subject)
ALTER TABLE student
ADD subject VARCHAR(20)
```

```
//delete a column(fee)
ALTER TABLE student
DROP COLUMN fee
```

```
//to change the datatype
ALTER TABLE student
MODIFY COLUMN grade VARCHAR(5)
```

```
//inserting values into specific fields
INSERT INTO student(id, name, grade)
VALUES(123, "BOB", 9)
```

```
//update a record on the table
UPDATE student
SET fee = fee*1.1
WHERE id = 123
//increase 123's fees by 10%
```

```
//delete a record(123's record)
DELETE FROM student
WHERE id = 123
```

```
//if you have two tables;Song,Artist
SELECT Song.SongTitle, Artist.ArtistName
FROM Song, Artist
WHERE (Song.ArtistID = Artist.ArtistID)
//using the .notation
```

```
//JOIN provides an alternative method of
//combining rows from 2 or more tables
SELECT Song.SongTitle, Artist.ArtistName
FROM Song
JOIN Artist
ON Song.ArtistID = Artist.ArtistID
//this and the statement above are the same
```

```
//like operator
SELECT name
FROM student
WHERE name LIKE 'B%';
//return names that start with B
```

```
//between..and
SELECT * //all
FROM student
WHERE age BETWEEN 14 AND 17
//returns all columns of which
//the students age is between 14 and 17
```

```
//not empty
SELECT grade
FROM student
WHERE fee IS NULL;
//returns all grades where the fee
//is null
```

```
//some useful functions
SELECT SUM(fee)
FROM student;
//will sum all the fees
```

```
SELECT COUNT(*)
FROM student
WHERE age < 17;
//counts the rows of all
//age less than 17
```