* Database operation (For MySQL)
  + status; //show all the information of the mysql (version) and database
  + Show databases;
  + Use *database*;
  + Show tables;
  + Desc *tableName*;
  + **CREATE TABLE** *table* (…, …, …, …)
    - **PRIMARY KEY** (…, ...),
    - **FOREIGN KEY** table1 (…, …) **REFERENCES** table2(…, …),
    - **NOT NULL**
    - **ON DELETE CASCADE**
    - **UNIQUE** (name, age), (p66)
    - **CONSTRAINT** StudentsKey PRIMARY KEY (sid), (p66)
  + **ALTER TABLE** *table* **ADD/DROP COLUMN |**  **MODIFY|CHANGE** *name* *type*;
    - AUTO\_INCREMENT
  + **DROP TABLE** <table>
  + **CREATE** [OR REPLACE] VIEW *XYZ*(Column1, Column2…) **AS**

SELECT A.Column1, A. Column2 ….

FROM ABC A …..

* **DROP VIEW** <view name>
* **INSERT INTO** table (…, …, …, …) **VALUES** (…, …, …, …);
* **UPDATE** table t

**SET** t.sid = 10000

**WHERE** t.sid = 5000

* Queries
  + SELECT [DISTINCT]

FROM

WHERE

* **LIKE**: %-- 0 or more char; \_ -- exactly one char
* Operation between temporary views:
  + **UNION**, **INTERSECT**, **EXCEPT**
* Operation in nested queries
  + **IN**/**NOT IN** (independent of outer query), **ANY**/**ALL**(P148)
  + **EXISTS**/**NOT EXISTS** (relates to outer query)
  + IN/NOT IN 相当于=ANY/<>ALL
* Aggregate operators
  + **COUNT** ([DISTINCT] A)
  + **SUM** ([DISTINCT] A)
  + **AVG** ([DISTINCT] A)
  + **MAX** (A)/MIN (A)
* **GROUP BY** and **HAVING**
  + SELECT [DISTINCE] select-list

FROM from-list

WHERE qualification

GROUP BY grouping-list

HAVING group-qualification

* Rules:
  + Every column that appears in select-list must also appear in grouping-list
  + A column appearing in the group-qualification must appear as the argument to an aggregation operator, or must also appear in grouping-list.
  + If GROUP BY is omitted, the entire table is regarded as a single group
* **EVERY** and **ANY** (P157)
* **AS** -- can be used to give name to a attribute or a temporary query result
* *rating* **IS NULL**/ **IS NOT NULL**
  + IS NULL/IS NOT NULL
* Outer Joins
  + **… LEFT/RIGHT/FULL OUTER JOIN** ... **ON**, **NATURAL LEFT OUTER JOIN**

Authors A LEFT OUTER JOIN BookAuthors BA ON A.NAME = BA.AUTHOR

Sailors S NATURAL LEFT OUTER JOIN Reserves R

* **ORDER BY** age **ASC**, rating **DESC**