MySQL **SQL**

* **in** DBMS, **data is** stored **in** a **file**

* within a **file**, the **rows are** stored sequentially

* **in** RDBMS, **table is not** a **file**; **every row is** a **file**

* **in** RDBMS, the **rows of** a **table are not** stored sequentially; the zows **of** a **table are** scattered **(**fragmented**)**
  all over the DB servez RD **when** you **INSERT** a **row into** a **table**, wherever it finds the free **space in** the DB server HD,
  it will store the **row** there

* the reason why RDBMS does this **is to** speed up the **INSERT statement (**considering multi**-user** environment**)**

* **in** a multi**-user** environment, **if** multiple users **are** inserting **rows** simultaneously **into** the same **table**,
  if the **rows** were **to** be stored sequentially, it would be very slow

* **when** you **SELECT from** a **table**, the **order of rows in** the **output** depends **on** the **row** address;
  it will always be **in** ascending **order of row** address

* **when** you **UPDATE** a **row**, **if** the **row length is** increasing **then** the **row** adress may change
  (only in case of Varchar length may increse **and** change**)**

  ==========

  Order by clause

  ==========