



MySQL

CSIS0521 MySQL 1



Assumption

- You have knowledge on DBMS
- You have knowledge on SQL (Structured Query Language)
- You have installed MySQL and have the MySQL server running on your system.
- Things in[] are optional in definition.

CSIS0521 MySQL 2



Using MySQL Command Line

- Simple commands:
 - use <database>;
 - show tables;
- Create a table:
 - create table users (
 - id varchar(16),
 - name(32),
 - studentno int(6),
 - phone char(8),
 - form char(1),
 - primary key (id)
- You can check the tables by
 - describe users

CSIS0521 MySQL 3



Creating a booking database

- create table booking (
 - bookid int(11) not null auto_increment,
 - booktime datetime,
 - userid varchar(16),
 - primary key (bookid)

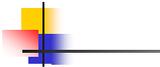
CSIS0521 MySQL 4



Column Data Types

- MySQL supports 3 column types: text, numeric and date.
- TEXT:
 - char (length)
 - varchar(length) – limited to 255 chars
 - enum(value1, value2, ..)
- NUMERIC
 - int (display_size) [unsigned][zerofill] – add zero on the left up to display size
 - double/float [(display_size[,decimals])] [zerofill]
 - decimal [(display_size[,decimals])[zerofill]

CSIS0521 MySQL 5



DATE type:

- date – YYYY-MM-DD format
- datetime – YYYY-MM-DD HH:MM:SS
- timestamp – automatically store the last date & time a record was updated.
- time – HH:MM:SS
- year[(2|4)] – store year in 2 digit format, from 1970-2069.

CSIS0521 MySQL 6

Column Attributes

- `auto_increment` : the first record will be stored with value 1, 2nd 2, etc. The column must be defined as unique or as a primary key.
- `default` – provide a default value
- `not null` – null value is not allowed
- `primary key` – define the column as primary keys.

CSIS0521

MySQL

7

Indexes and Primary Keys

- MySQL can create indexes on columns to speed up access.
- Use the following in create table command:
index <index_name> (<column_name>)
- Supplying a name of the index is not mandatory.
- You can also create a primary key, which is a special type of index, acting as a unique key for each row in the table.
- Primary keys need to be unique, you can use an `auto_increment` column as a key.

CSIS0521

MySQL

8

Altering Tables

- **Changing table name**
`alter table <table_name> rename <new_table_name>;`
- **Adding columns**
`alter table <table_name> add column <column_name> <column_type>
<column parameters>`
 - Column parameters define where to add the column:
first, last, before <col_name>, after <col_name>.
- **Changing column definition**
`alter table <table_name> change <old_column_name>
<new_column_name><column_attributes>`
 - You can specify the same name for old and new column name, if you just want to change the definition of the column, or use `MODIFY`.

CSIS0521

MySQL

9

- **Alter table <table_name> modify <column_name>
<column_type><column attributes>;**
- **Dropping columns – remove a column**
`alter table <table_name> drop column <column_name>;`
- **Changing primary keys and indexes:**
`alter table drop index <index_name>;`
`alter table add index <index_name> (column_name);`
`alter table add primary key (<column_name>);`
- **Dropping tables:**
`drop table <table_name>;`

CSIS0521

MySQL

10

Other information

- Show databases;
`show tables;`
`describe <table_name>;`
`show index from <table_name>;`
`show create table <table_name>;`
show the command for creating that table, can be used to duplicate a table.

CSIS0521

MySQL

11

MySQL Queries - insert

- The insert query enables you to add data to an existing table.
`insert into <table_name>
(<column_list>) values (<value_list>);`
- E.g.
`insert into users (id,name,studentno,phone)
values ('kpchan', 'Kwok-Ping Chan', 123456,
'28592185');`
- You can add more rows into the database using the insert command.

CSIS0521

MySQL

12



- Strings in MySQL are enclosed using ' '.
- You can use \' if data contain single quote.
- MySQL also support "", but SQL standard uses "".
- The order you supplies the column names can be in any order.
- Alternatively, you can use:
insert into <table> set <column1>=<value1>,
<column2>=<value2>, ...;
- If values not specified, NULL values will be assumed.

CSIS0521 MySQL 13



MySQL Query - select

- Select data from the database and display or parse it accordingly.
- Basic form:
select <column_list> from <table_name>;
e.g. select id, names from users;
select * from users;
- You can select with restriction
select <column_list> from <table_name>
where <condition>;
e.g. select id,name from users where form=5;

CSIS0521 MySQL 14



Operators in where

- =, != (<>), >, <, >=, <=
- Can use a between operator:
- E.g. Select id, name from users where form between 1 and 3;
- Using wildcard chars with like:
% - matches 0 or more chars
_ - matches one char
- E.g. select id,name from users where name like 'd%';

CSIS0521 MySQL 15



- You can use AND, OR etc to construct more complex query:
 - Select id, name from users where name like 'd%' and form=3;
- You can order the query result according to a particular column, default is ascending, or desc for descending order:
 - Select id,name from users where name like 'd%' order by studentno desc;
- You can add a limit to limit the number of rows to be displayed:
 - Select id,name from users where name like 'd%' order by studentno desc limit 5;

CSIS0521 MySQL 16



Selecting from multiple tables

- For example, you want to check all bookings from students in form 3.
- First, find all students in form 3:
select id from users where form=3;
- From the result, then select one by one
select date from booking where userid=' ';
to find out all bookings from all students in form 3.

CSIS0521 MySQL 17



- You can use
select booking.date from booking, users
where users.form=3 and
users.id=booking.userid;
- Use "." to define the fields from a particular table, e.g. booking.date and the date field from table booking.
- You may have same column names in different tables
- Joins in database

CSIS0521 MySQL 18



Inner Join

- Select data based on two columns, each from a separate table, being equal.
- MySQL supports using inner join keyword
- Select booking.date, booking.userid from booking inner join users on booking.userid=users.id where users.id=kpchan

CSIS0521 MySQL 19



Left Outer Join

- If the on clause matches 0 records in the 2nd table, a row in the result will still be returned but with NULL values for each column for the 2nd table.
- E.g. select booking.date, booking.userid from booking left outer join users on booking.userid=users.id;
- Note that if a booking.userid if not found in users, the row is still returned, with NULL entries in the columns of users.

CSIS0521 MySQL 20



Counting result sets

- Use count() function to count the number of rows in the result set.
- select count(*) from users.
- Select form, count(*) from users where form=7 group by form;

CSIS0521 MySQL 21



Deleting records

- delete from <table_name> [where <condition>];
- Condition is optional, it means you will delete all.
- E.g. delete from users where form=7;
- Note that delete cannot be undone!

CSIS0521 MySQL 22



Updating records

- To update a record, use:
update <table_name> set
<colname>=<value>,
<colname>=<value>, ...
where <condition>
- E.g. update users set form=7 where form=6;
- Update users set form=form+1 where form!=7;

CSIS0521 MySQL 23



Backing up/Restoring MySQL

- Use the program mysqldump to dump the database
- mysqldump -u root -p test will dump the database test
- The output is commands that can be used to restore the database.
- You can save it to a file using output redirection.

CSIS0521 MySQL 24



- To restore, use `mysql -u root -p < dumpfile` on the command line.

CSIS0521 MySQL 25



Designing your database

- Some guideline for database design
 - Think about the real world objects you are modeling.
 - Each class of real-world object will require its own table, e.g. users and booking in previous examples.
 - If you have different courts, then you also need another database for the courts.

CSIS0521 MySQL 26



- Avoid storing redundant data
 - Why don't we store the users info in the same table as booking?
 - We have to store this information multiple times.
 - May lead to update anomalies.
 - Modification anomalies
 - Insertion anomalies
 - Deletion anomalies (e.g. loss of customer information once all rows involving that customers are deleted)

CSIS0521 MySQL 27



- Use Atomic Column Values
 - Each attribute in each row store only one thing.
 - For example, if a customer ordered multiple items, do not store all items in one single row.
 - It would be difficult for further processing.
 - E.g. a customers order two books, and the following is stored:

orderid	custid	amount	date	books
1	3	102.5	1/10/06	Book1,Book2
2	4	32.6	11/11/06	Book2, Book3
 - Then querying the number of copies of Book2 sold can be very troublesome.

CSIS0521 MySQL 28



- Choose Sensible Keys
 - Keys chosen should be unique
 - E.g. unique user-id, cust-id, Book-id (ISBN) etc.
- Think about the Questions you want to ask the database
 - Make sure database contains all data required.
 - Appropriate link exists between tables

CSIS0521 MySQL 29



- Avoid designs with many NULL attributes
 - Waste storage
 - When a user sees a NULL in a table, they don't know if the value is irrelevant, or there is a mistake, or data not yet entered.
 - Use alternate design, e.g. add additional tables.
 - E.g. a book company want to add a book review section to book entries. Since this book review may be sparse, try to use a separate table for book reviews, rather than adding a column for every records.

CSIS0521 MySQL 30



First Normal Form

- Database are in 1NF if it observes the following rules:
 - Ensure that each table has a primary key
 - Eliminate repeating groups
 - Each attribute must contain a single value, not a set of values.
- In the book order example, we have to separate each book order into different rows.

CSI50521 MySQL 31



Second Normal Form

- The database meet all requirements of the 1NF
- Data which is redundantly duplicated across multiple rows of a table is moved out to a separate table.
- E.g. for a book database, you can move the author, publishers to a different table.

CSI50521 MySQL 32



Third Normal Form

- 3NF requires that data stored in a table be dependent only on the primary key, and not on any other field in the table.
 - The database must meet all the requirements of 2NF
 - Any field that is dependent not only on the primary key but also on another field is moved out to a separate table.

CSI50521 MySQL 33