Fundamentals of Database Systems
[SQL – I]

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
Indian Statistical Institute, Kolkata

August, 2019

1 Preliminaries

2 Data Definition
- Principle Structure
- Database Creation
- Database Modification

3 Problems

Basics of SQL

SQL or structured query language is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS). SQL uses a combination of relational algebra and relational calculus constructs.

SQL is not only for querying, rather it also helps in defining the structure of the data, modifying the data and specifying the security constraints.

Basics of SQL

SQL or structured query language is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS). SQL uses a combination of relational algebra and relational calculus constructs.

SQL is not only for querying, rather it also helps in defining the structure of the data, modifying the data and specifying the security constraints.

Note: The SQL keywords are case-insensitive, however, they are often written in uppercase. In some setups, table and column names are case-sensitive.

SQL functionalities

- **Data-definition language (DDL)** – provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
- **Data-manipulation language (DML)** – includes commands to work on attributes, insert tuples into, delete tuples from, and modify tuples in the database.
- **View definition** – includes commands for defining views.
- **Transaction control** – includes commands for specifying the beginning and ending of transactions.
- **Embedded SQL and dynamic SQL** – embeds SQL statements into general-purpose programming languages.
- **Integrity** – includes commands for specifying integrity constraints that the data stored in the database must satisfy.
- **Authorization** – includes commands for specifying access rights to relations and views.

History

“An SQL query goes into a bar, walks up to two tables and asks, ‘May I join you?’.” – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.

History

“An SQL query goes into a bar, walks up to two tables and asks, ‘May I join you?’.” – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.
1986: American national Standards Institute (ANSI) and International Organization for Standardization (ISO) published an SQL standard SQL-86.

History

“An SQL query goes into a bar, walks up to two tables and asks, ‘May I join you?’.” – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.
1986: American national Standards Institute (ANSI) and International Organization for Standardization (ISO) published an SQL standard SQL-86.
1987: IBM published its own corporate SQL standard Systems Application Architecture Database Interface (SAA-SQL).

History

“An SQL query goes into a bar, walks up to two tables and asks, ‘May I join you?’.” – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.
1986: American national Standards Institute (ANSI) and International Organization for Standardization (ISO) published an SQL standard SQL-86.
1987: IBM published its own corporate SQL standard Systems Application Architecture Database Interface (SAA-SQL).
1989: ANSI published an extended version SQL-89.

History

"An SQL query goes into a bar, walks up to two tables and asks, 'May I join you?'." – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.
1986: American national Standards Institute (ANSI) and International Organization for Standardization (ISO) published an SQL standard SQL-86.
1987: IBM published its own corporate SQL standard Systems Application Architecture Database Interface (SAA-SQL).
1989: ANSI published an extended version SQL-89.
1992: A major extended version SQL-92 was published.

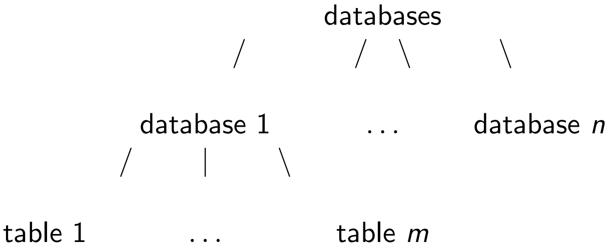
History

"An SQL query goes into a bar, walks up to two tables and asks, 'May I join you?'." – Anonymous.

1970s: Original version called Sequel, developed as a part of the System R project, was first implemented by IBM.
1986: American national Standards Institute (ANSI) and International Organization for Standardization (ISO) published an SQL standard SQL-86.
1987: IBM published its own corporate SQL standard Systems Application Architecture Database Interface (SAA-SQL).
1989: ANSI published an extended version SQL-89.
1992: A major extended version SQL-92 was published.
1999-2016: The versions SQL:1999, SQL:2003, SQL:2006, SQL:2008, SQL:2011 and SQL:2016 were published.

Data view through SQL

In practice, the databases (as a whole) comprises several separate database and each database consists of several tables.



databases

database 1 . . . database *n*

table 1 . . . table *m*

Note: The MySQL Community Server can be downloaded from `https://dev.mysql.com/downloads/mysql`.

Connecting with MySQL

```
 $ mysql -u <user_name> -p
Enter password:
mysql> _
mysql> show databases;
mysql> connect <database_name>;
mysql> show tables;
mysql> desc <table_name>;
mysql> exit
$ _
```

We will be using the accounts with username/password `root/mysql` (DBA) and `student/student123` (general user) on MySQL during the course.

Note: In Oracle SQL, `scott` and `tiger` are the general username and password, respectively. The default password for the DBA is `system`.

Principle structure of defining a table

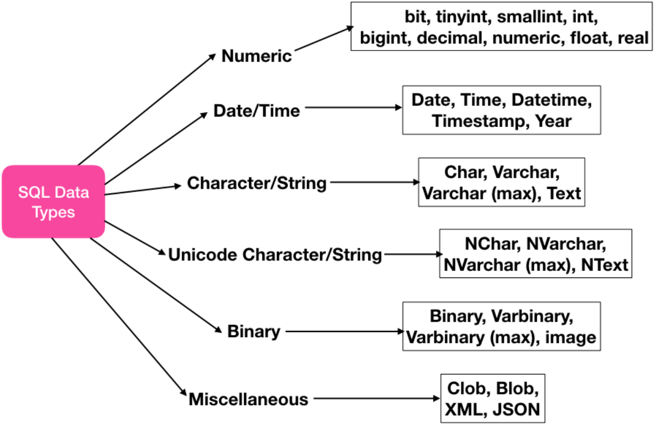
A typical SQL query for data definition appears as follows:

`create table` $R($
$\mathtt{A}_1 D_1, A_2 D_2, \ldots, A_k D_k$,
$(IC_1), \ldots, (IC_l)$
);

Here, each A_i represents an attribute in the schema of relation R, each D_i denotes the data type of values in the domain of the corresponding attribute A_i, and IC_i symbolizes an integrity-constraint.

Note: SQL is a freeform language.

The data types in SQL


SQL Data Types
Numeric
bit, tinyint, smallint, int, bigint, decimal, numeric, float, real
Date/Time
Date, Time, Datetime, Timestamp, Year
Character/String
Char, Varchar, Varchar (max), Text
Unicode Character/String
NChar, NVarchar, NVarchar (max), NText
Binary
Binary, Varbinary, Varbinary (max), image
Miscellaneous
Clob, Blob, XML, JSON

The data types in MySQL – Numerics

Type	Size (bits)	Minimum	Maximum
tinyint(dig)	8	-2^7	$2^7 - 1$
unsigned tinyint(dig)	8	0	$2^8 - 1$
smallint(dig)	16	-2^{15}	$2^{15} - 1$
unsigned smallint(dig)	16	0	$2^{16} - 1$
mediumint(dig)	24	-2^{23}	$2^{23} - 1$
unsigned mediumint(dig)	24	0	$2^{24} - 1$
int(dig)	32	-2^{31}	$2^{31} - 1$
unsigned int(dig)	32	0	$2^{32} - 1$
bigint(dig)	64	-2^{63}	$2^{63} - 1$
unsigned bigint(dig)	64	0	$2^{64} - 1$
real(dig, dec)	32	$-3.40E + 38$	$3.40E + 38$
float(dig, dec)	64	$-1.79E + 308$	$1.79E + 308$
decimal(dig, dec)	136	$-10^{38} + 1$	$10^{38} - 1$
numeric(dig, dec)	136	$-10^{38} + 1$	$10^{38} - 1$

Note: The maximum number of digits before and after the decimal point can be specified with *dig* and *dec*, respectively.

The data types in MySQL – Text

Type	Storage	Maximum
char(size)	Fixed length string	255 characters
varchar(size)	Variable length string	255 characters
tinytext(size)	Character large object/CLOB	255 characters
text(size)	Character large object/CLOB	65,535 characters
mediumtext(size)	Character large object/CLOB	16,777,215 characters
longtext(size)	Character large object/CLOB	4,294,967,295 characters
blob(size)	Binary large object/BLOB	65,535 bytes
mediumblob(size)	Binary large object/BLOB	16,777,215 bytes
longblob(size)	Binary large object/BLOB	4,294,967,295 bytes
enum(x,y,z,etc.)	A list of possible values	65535 values

Note: A string can contain letters, numbers, and special characters. The maximum number of characters in a string can be specified with *size*.

Consider a table

Table: IPL

YEAR	VENUE	WINNER	PoS
8	India	Rajasthan Royals	Shane Watson
9	South Africa	Deccan Chargers	Adam Gilchrist
10	India	Chennai Super Kings	Sachin Tendulkar
11	India	Chennai Super Kings	Chris Gayle
12	India	Kolkata Knight Riders	Sunil Narine
13	India	Mumbai Indians	Shane Watson
14	India, UAE	Kolkata Knight Riders	Glenn Maxwell
15	India	Mumbai Indians	Andre Russell
16	India	Sunrisers Hyderabad	Virat Kohli
17	India	Mumbai Indians	Ben Stokes
18	India	Chennai Super Kings	Sunil Narine
19	India	Mumbai Indians	Andre Russell

Creating a table

The IPL table, on which we will be working on further, can be created with the following SQL query.

```
 create table IPL(
YEAR tinyint(4) not null,
VENUE char(50),
WINNER char(30),
PoS char(30),
primary key (YEAR)
);
```

<u>Note</u>: The attribute `YEAR`, which cannot be `null`, is defined as the primary key of IPL table.

Table creation with ease

Try this out!!!

SQLizer – Easily convert files into SQL databases

`https://sqlizer.io`

Deleting a table

The IPL table can be deleted from database using the following SQL query.

```
drop table IPL;
```

Altering a table

The IPL table can be altered by adding a new attribute A_p and mentioning its domain D_p (data type) as follows. All the existing tuples will be assigned a `null` value for the new attribute A_p.

`alter table IPL add` $A_p D_p$;

Altering a table

The IPL table can be altered by adding a new attribute A_p and mentioning its domain D_p (data type) as follows. All the existing tuples will be assigned a `null` value for the new attribute A_p.

```
 alter table IPL add A_p D_p;
```

The IPL table can be altered by dropping an existing attribute A_p as follows.

```
alter table IPL drop A_q;
```

Altering a table

The IPL table can be altered by adding a new attribute A_p and mentioning its domain D_p (data type) as follows. All the existing tuples will be assigned a `null` value for the new attribute A_p.

`alter table IPL add` $A_p D_p$`;`

The IPL table can be altered by dropping an existing attribute A_p as follows.

`alter table IPL drop` A_q`;`

Note: New primary keys can also be added/dropped in a similar way.

Renaming a table and its attributes

The IPL table and its attributes can be renamed and reused, as and when required, within an SQL query as follows:

```
 select IPL1.PoS from IPL as IPL1, IPL as IPL2 where
IPL1.PoS = IPL2.PoS and IPL1.YEAR > IPL2.YEAR and
IPL2.WINNER = ‘Rajasthan Royals’;
```

Renaming a table and its attributes

The IPL table and its attributes can be renamed and reused, as and when required, within an SQL query as follows:

```
 select IPL1.PoS from IPL as IPL1, IPL as IPL2 where
IPL1.PoS = IPL2.PoS and IPL1.YEAR > IPL2.YEAR and
IPL2.WINNER = ‘Rajasthan Royals’;
```

This will yield the names of Player of Series (PoS) winners while playing for ‘Mumbai Indians’ earlier and then winning again at a later time.

Delete operation on the tuples

```
 delete from IPL
where YEAR < 10;
```

Note: It works on the entire tuple and can not delete values on arbitrary attributes.

Delete operation on the tuples

Table: IPL

YEAR	VENUE	WINNER	PoS
10	India	Chennai Super Kings	Sachin Tendulkar
11	India	Chennai Super Kings	Chris Gayle
12	India	Kolkata Knight Riders	Sunil Narine
13	India	Mumbai Indians	Shane Watson
14	India, UAE	Kolkata Knight Riders	Glenn Maxwell
15	India	Mumbai Indians	Andre Russell
16	India	Sunrisers Hyderabad	Virat Kohli
17	India	Mumbai Indians	Ben Stokes
18	India	Chennai Super Kings	Sunil Narine
19	India	Mumbai Indians	Andre Russell

Insert operation

```
 insert into IPL
values (8, ‘India’, ‘Rajasthan Royals’,‘Shane
Waton’);
```

```
insert into IPL values (9, ‘South Africa’, ‘Deccan
Chargers’,‘Adam Gilchrist’);
```

Note: You can optionally mention the attributes as well as follows “`insert into IPL (YEAR, VENUE, WINNER, PoS) values (9, ‘South Africa’, ‘Deccan Chargers’,‘Adam Gilchrist’);`”.

Insert operation

Table: IPL

YEAR	VENUE	WINNER	PoS
8	India	Rajasthan Royals	Shane Watson
9	South Africa	Deccan Chargers	Adam Gilchrist
10	India	Chennai Super Kings	Sachin Tendulkar
11	India	Chennai Super Kings	Chris Gayle
12	India	Kolkata Knight Riders	Sunil Narine
13	India	Mumbai Indians	Shane Watson
14	India, UAE	Kolkata Knight Riders	Glenn Maxwell
15	India	Mumbai Indians	Andre Russell
16	India	Sunrisers Hyderabad	Virat Kohli
17	India	Mumbai Indians	Ben Stokes
18	India	Chennai Super Kings	Sunil Narine
19	India	Mumbai Indians	Andre Russell

Note: The inserted tuples are added as per the order of primary key value, otherwise (no primary key) at the bottom.

Update operation

```
 update IPL
set YEAR = YEAR + 2000
where YEAR < 2000;
```

Update operation

Table: IPL

YEAR	VENUE	WINNER	PoS
2008	India	Rajasthan Royals	Shane Watson
2009	South Africa	Deccan Chargers	Adam Gilchrist
2010	India	Chennai Super Kings	Sachin Tendulkar
2011	India	Chennai Super Kings	Chris Gayle
2012	India	Kolkata Knight Riders	Sunil Narine
2013	India	Mumbai Indians	Shane Watson
2014	India, UAE	Kolkata Knight Riders	Glenn Maxwell
2015	India	Mumbai Indians	Andre Russell
2016	India	Sunrisers Hyderabad	Virat Kohli
2017	India	Mumbai Indians	Ben Stokes
2018	India	Chennai Super Kings	Sunil Narine
2019	India	Mumbai Indians	Andre Russell

Understanding the concepts in a better way

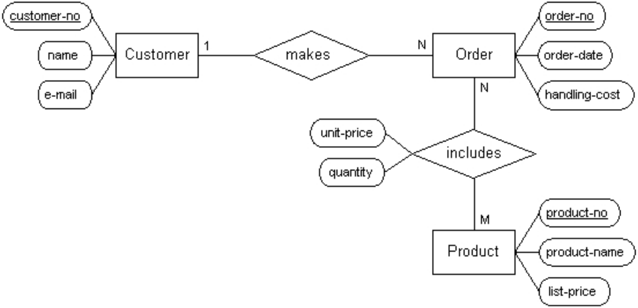
Try this out!!!

RAT – Relational Algebra Translator

`http://www.slinfo.una.ac.cr/rat/rat.html`

Problems

1. Consider the following E-R diagram representing the details about products, customers, and orders that were placed by the customers of Amazon Freedom Sale during August 8-11, 2019. Create the database and insert data.


customer-no
name
e-mail
Customer
1
makes
N
Order
order-no
order-date
handling-cost
N
unit-price
quantity
includes
M
Product
product-no
product-name
list-price