



METIS

SQL FOUNDATIONS

INTRODUCTION TO DATABASES & SQL

METIS



Data is Everywhere

We produce more than 2.5 quintillion bytes of data each day, and 90% of the world's data was created in the last 2 years.

Making practical use of that data, however, often requires data organization coupled with programming skills.



What is a database (DB)?

Collection of organized data stored within a computer

Relational Database

- Database structured as table(s)
- Clear information relationships
- Data types dictated in schema
- Structured Query Language (SQL)

Non-Relational (NoSQL) Database

- DB does not follow the relational model
- Allows more data types
- Allows unstructured data
- Preferred for very large datasets
- MongoDB popular option



Relational Database Example

Relational table detailing activities offered by Company XYZ below

HEADER →

Each row describes one item →

Each column describes one attribute ↓

id	item	activity_level	category	family_friendly
1	wind surfing	4	sport	0
2	walk on Great Wall of China	2	site seeing	1
3	climb Mount Everest	5	sport	0
4	French cuisine package	0	food and beverage	1
5	geocaching package	1	sport	1
6	Broadway musical experience	0	culture	1
7	helicopter over Grand Canyon	1	site seeing	1
8	attend Cannes Film Festival	0	culture	0
9	Napa Valley wine tasting	0	food and beverage	0
10	Alaskan glacier cruise	2	site seeing	1



Relational Database Example

Activities Table

id	item	activity_level	category	family_friendly
1	wind surfing	4	sport	0
2	walk on Great Wall of China	2	site seeing	1
3	climb Mount Everest	5	sport	0
4	French cuisine package	0	food and beverage	1
5	geocaching package	1	sport	1
6	Broadway musical experience	0	culture	1

Collection of associated tables make up database

Vendors Table

id	vendor_name	phone_number	city	activity_id	price
1	Phil's Surfing Emporium	800-345-SURF	Honolulu	1	250.0000
2	Fun, Sun, and Surfing	888-541-1219	San Diego	1	300.0000
3	Trekking Everest	800-212-1001	Portland	3	14500.0000
4	Le Meilleur de la Mer	212-905-5521	New York	4	275.0000
5	Jacques et Lise	415-555-1000	San Francisco	4	199.9900
6	Live Ticket New York	347-333-SHOW	New York	6	350.0000

What is SQL?



- Structured Query Language
- Programming language for interacting with relational databases
- SQL developed in 1970s at IBM
- Series of commands to create, query, and alter databases
- Database Management Systems: SQL Server, Oracle, IBM Db2

SQL Flavors



- Each DBMS has slightly different syntax for interacting with DB
- Example: SQL Server is not case sensitive; PostgreSQL is
- Once you learn one SQL dialect, others are easier to master
- In this course we will use **Microsoft SQL Server**
- Interface: **Microsoft SQL Server Management Studio (SSMS)**

CREATING A TABLE

METIS



Creating a Table

1. CREATE TABLE command
2. Name the table
3. Column (attribute) names
4. Column data types

```
CREATE TABLE inventory (
    id INTEGER PRIMARY KEY,
    name TEXT,
    quantity INTEGER
);
```

id	name	quantity
----	------	----------

Schema of a Table: Constraints



```
CREATE TABLE inventory (
    id INTEGER PRIMARY KEY,
    name TEXT,
    quantity INTEGER
);
```

1. Data types constraint (INTEGER, TEXT): wrong type will produce an error
2. Another constraint: PRIMARY KEY—field that uniquely identifies each row



Inserting Data into a Table

```
INSERT INTO inventory VALUES (1, 'tiger t-shirt', 10);
```

Table Name

Row value for each column: id, name, quantity

Resulting data table once CREATE TABLE and INSERT INTO commands executed:

id	name	quantity
1	tiger t-shirt	10

Inserting Data into a Table



Continue with INSERT INTO statements for each item in inventory log:

```
INSERT INTO inventory VALUES (1, 'tiger t-shirt', 10);  
INSERT INTO inventory VALUES (2, 'giraffe-print bag', 18);  
INSERT INTO inventory VALUES (3, 'elephant tie', 15);  
INSERT INTO inventory VALUES (4, 'zebra-striped pants', 7);
```



Note: Because the 'id' column is the primary key, these values must be unique!



Our First SQL Table

id	name	quantity
1	tiger t-shirt	10
2	giraffe-print bag	18
3	elephant tie	15
4	zebra-striped pants	7

```
CREATE TABLE inventory (
    id INTEGER PRIMARY KEY,
    name TEXT,
    quantity INTEGER
);
```

```
INSERT INTO inventory VALUES (1, 'tiger t-shirt', 10);
INSERT INTO inventory VALUES (2, 'giraffe-print bag', 18);
INSERT INTO inventory VALUES (3, 'elephant tie', 15);
INSERT INTO inventory VALUES (4, 'zebra-striped pants', 7);
```

DATA TYPES

METIS

Creating a Table (Revisited)



Returning to the inventory example, three columns containing integers or text:

```
CREATE TABLE inventory (
    id INTEGER PRIMARY KEY,
    name TEXT,
    quantity INTEGER
);
```



Data Types List (SQL Server)

BIGINT

BIT

DECIMAL

INTEGER (INT)

MONEY

NUMERIC

SMALLINT

SMALLMONEY

TINYINT

FLOAT

REAL

DATE

DATETIME

DATETIME2

DATETIMEOFFSET

SMALLDATETIME

TIME

CHAR

TEXT

VARCHAR

NCHAR

NVARCHAR

BINARY

IMAGE

VARBINARY

+ more...



Creating Table with More Data Types

Induce three new columns in inventory table for tracking vendor unit price, date of last shipment, and whether or not we should reorder:

```
CREATE TABLE inventory (
    id INTEGER PRIMARY KEY,
    name VARCHAR(100),
    quantity INTEGER,
    vendor_unit_price MONEY,
    last_shipment DATE,
    reorder BIT
);
```

Switched from TEXT
to VARCHAR(100);
No more than 100
characters permitted



Creating Table with More Data Types

Now `INSERT INTO` commands contain information about unit price, last shipment, and reordering:

```
INSERT INTO inventory VALUES (1, 'tiger t-shirt', 10, 4.25,  
'2018-01-22', 'TRUE');  
INSERT INTO inventory VALUES (2, 'giraffe-print bag', 18, 24.99,  
'2018-02-26', 'FALSE');  
INSERT INTO inventory VALUES (3, 'elephant tie', 15, 13.19,  
'2018-02-26', 'FALSE');  
INSERT INTO inventory VALUES (4, 'zebra-striped pants', 7,  
16.88, '2018-01-08', 'TRUE');
```



Creating Table with More Data Types

Resulting table created with new data types:

id	name	quantity	vendor_unit_price	last_shipment	reorder
1	tiger t-shirt	10	4.2500	2018-01-22	1
2	giraffe-print bag	18	24.9900	2018-02-26	0
3	elephant tie	15	13.1900	2018-02-26	0
4	zebra-striped pants	7	16.8800	2018-01-08	1



Note: 'TRUE' and 'FALSE' are converted to 1 and 0, respectively, since they are BIT data types



Creating Table with Missing Values

Add two more rows with some information missing:

```
INSERT INTO inventory (id, name, quantity, reorder) VALUES (
    5, 'peacock feather hat', 2, 'FALSE');

INSERT INTO inventory (id, name, vendor_unit_price) VALUES (
    6, 'leopard-print scarf', 8.55);
```

Note: Now must include column names after table name so SQL knows which attributes have values.



Creating Another Table

SQL inserts NULL for any missing value:

id	name	quantity	vendor_unit_price	last_shipment	reorder
1	tiger t-shirt	10	4.2500	2018-01-22	1
2	giraffe-print bag	18	24.9900	2018-02-26	0
3	elephant tie	15	13.1900	2018-02-26	0
4	zebra-striped pants	7	16.8800	2018-01-08	1
5	peacock feather hat	2	NULL	NULL	0
6	leopard-print scarf	NULL	8.5500	NULL	NULL

EXERCISE: TABLES AND DATA TYPES

METIS

Tables and Data Types



- Next, we will create two tables: *activities*, and *vendors*
- Manually set up the tables in the next two slides.



Optional Exercise: Tables and Data Types

- Recreate the following activities table in SQL to log the items offered by Company XYZ.

id	item	activity_level	category	family_friendly
1	wind surfing	4	sport	0
2	walk on Great Wall of China	2	site seeing	1
3	climb Mount Everest	5	sport	0
4	French cuisine package	0	food and beverage	1
5	geocaching package	1	sport	1
6	Broadway musical experience	0	culture	1

Optional Exercise: Tables and Data Types



- Create another table for the company's vendors as seen below. Again think about each data type and what will be the primary key for this table.

id	vendor_name	phone_number	city	activity_id	price
1	Phil's Surfing Emporium	800-345-SURF	Honolulu	1	250.0000
2	Fun, Sun, and Surfing	888-541-1219	San Diego	1	300.0000
3	Trekking Everest	800-212-1001	Portland	3	14500.0000
4	Le Meilleur de la Mer	212-905-5521	New York	4	275.0000
5	Jacques et Lise	415-555-1000	San Francisco	4	199.9900
6	Live Ticket New York	347-333-SHOW	New York	6	350.0000