Introduction to SQL

Prof. Hyuk-Yoon Kwon

https://sites.google.com/view/seoultech-bigdata

Most parts are based on slides used in Stanford (http://web.stanford.edu/class/cs145)

Contents

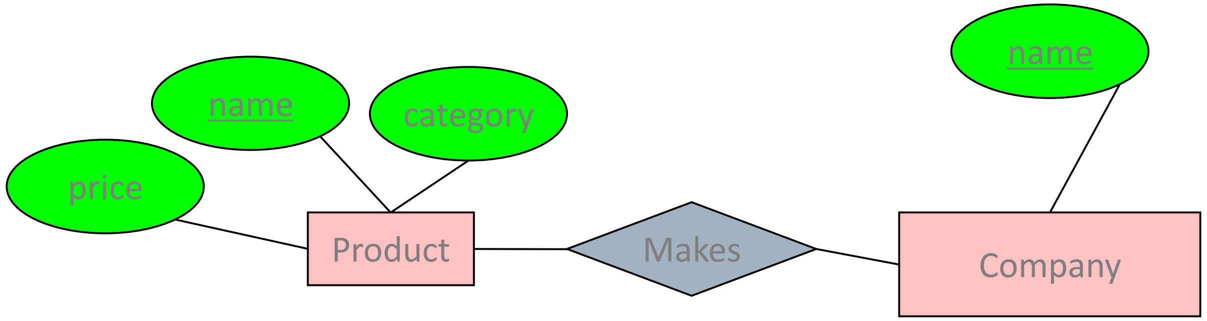
- **Summary of Previous Lecture: ER Model**
- **Today's lecture:**
  - Advanced ER Concepts
  - Introduction to SQL
    1. SQL introduction & schema definitions
    2. Basic single-table queries
    3. Multi-table queries

Database Design Process

1. Requirements Analysis
2. Conceptual Design
3. Logical, Physical, Security, etc.

E/R Model & Diagrams used

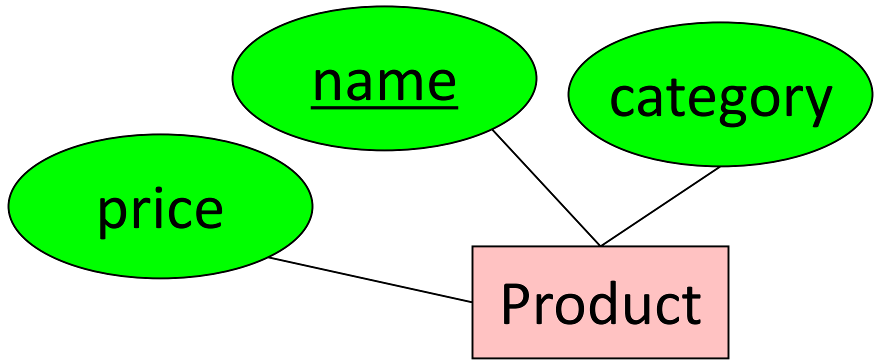
This process is iterated **many** times



E/R is a *visual syntax* for DB design which is ***precise enough*** for technical points, but ***abstracted enough*** for non-technical people

Entities and Entity Sets

- **An entity set has attributes**
  - Represented by ovals attached to an entity set

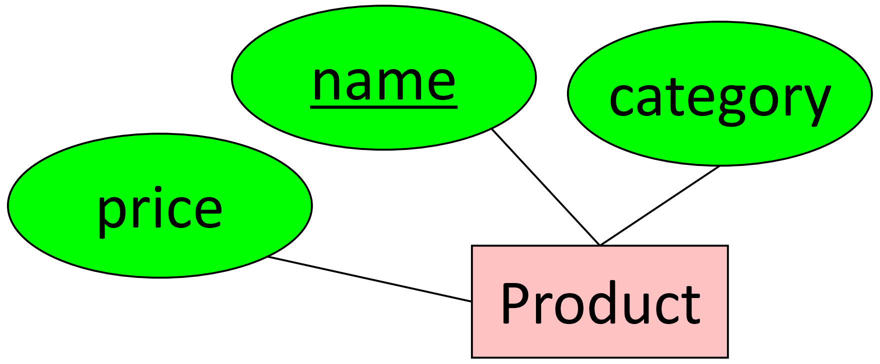


Shapes **are** important.
Colors **are not.**

Keys

- A ***key*** **is a minimal set of attributes that uniquely identifies an entity.**

Denote elements of the primary key by underlining.



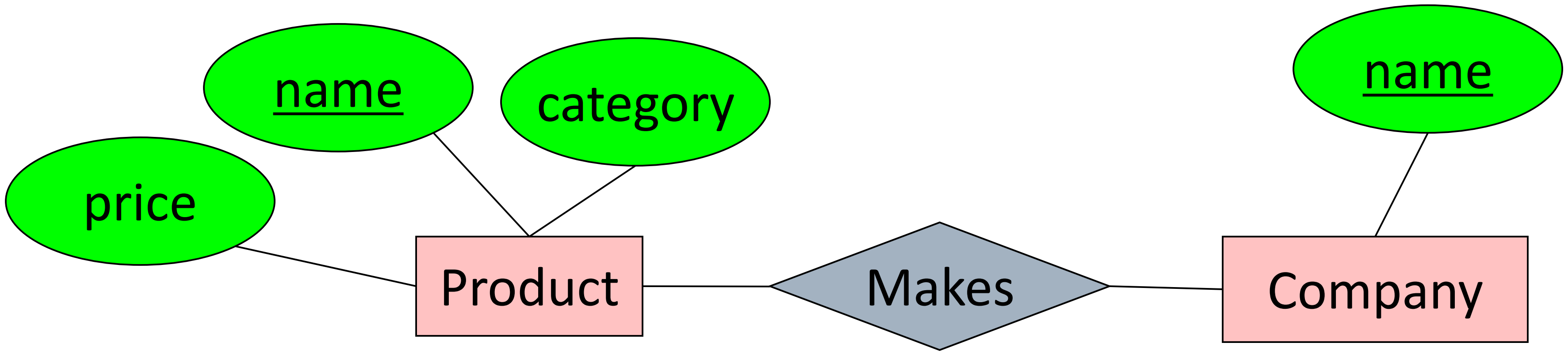
Here, {name, category} is **not** a key (it is not *minimal*).

If it were, what would it mean?

The E/R model forces us to designate a single **primary** key, though there may be multiple candidate keys

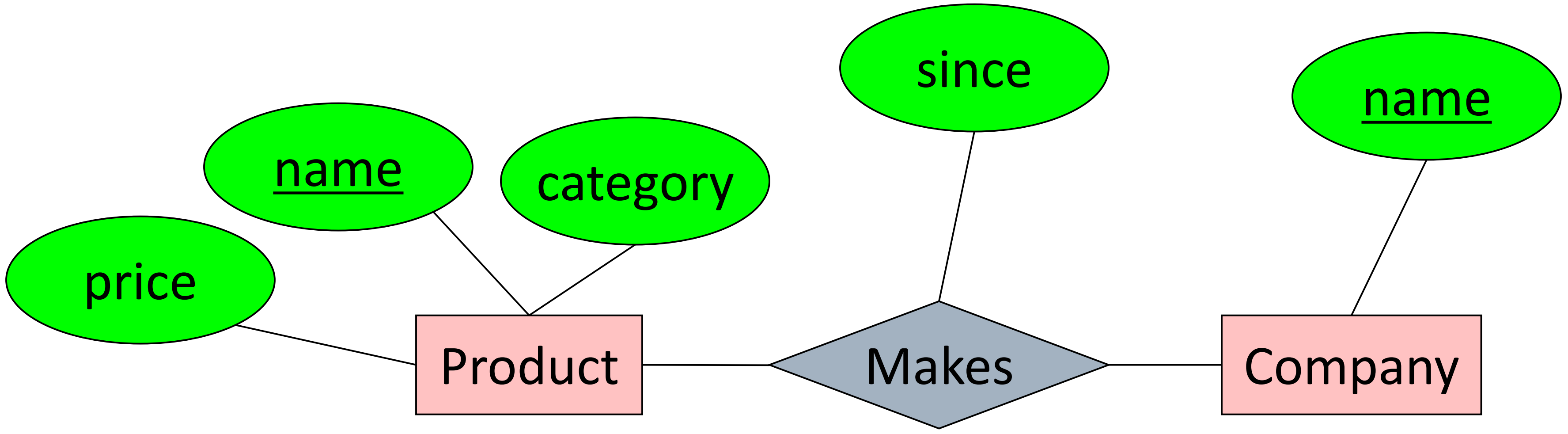
The R in E/R: Relationships

- **A relationship is between two entities**



Relationships and Attributes

- **Relationships may have attributes as well.**



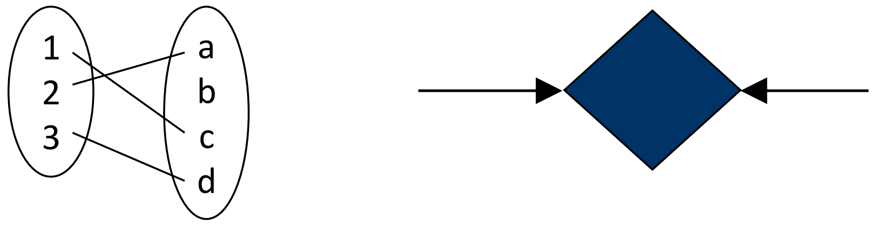
For example: "since" records when company started making a product

Note: "*since*" is implicitly unique per pair here! Why?

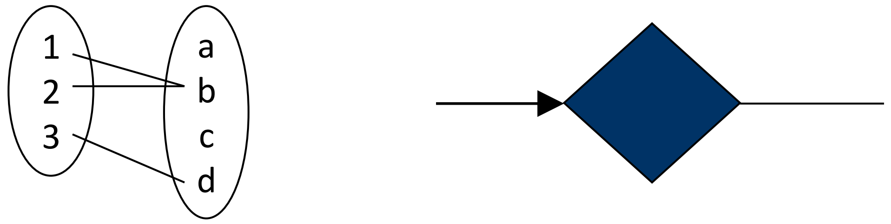
Note #2: Why not "how long"?

Multiplicity of E/R Relationships

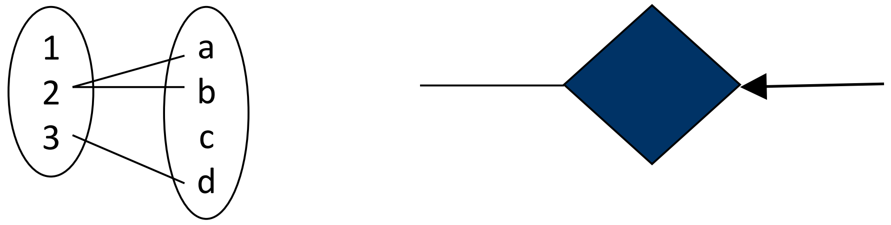
One-to-one:



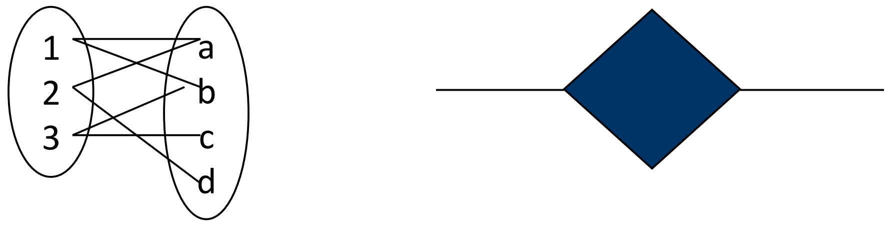
Many-to-one:



One-to-many:



Many-to-many:



Indicated using arrows

X -> Y means **<u>there exists a function mapping from X to Y</u>** (*recall the definition of a function)*

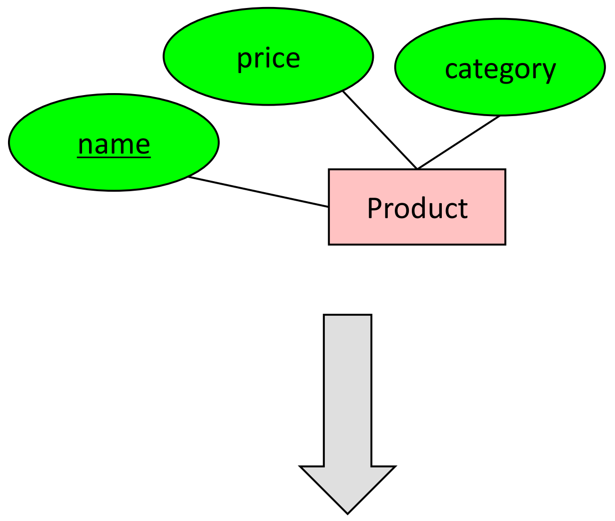
From E/R Diagrams to Relational Schema

Key concept:

Both ***Entity sets*** and ***Relationships*** become relations (tabl
es in RDBMS)

From E/R Diagrams to Relational Schema

- An entity set becomes a relation (multiset of tuples / table)
  - Each tuple is one entity
  - Each tuple is composed of the entity's attributes, and has the same primary key

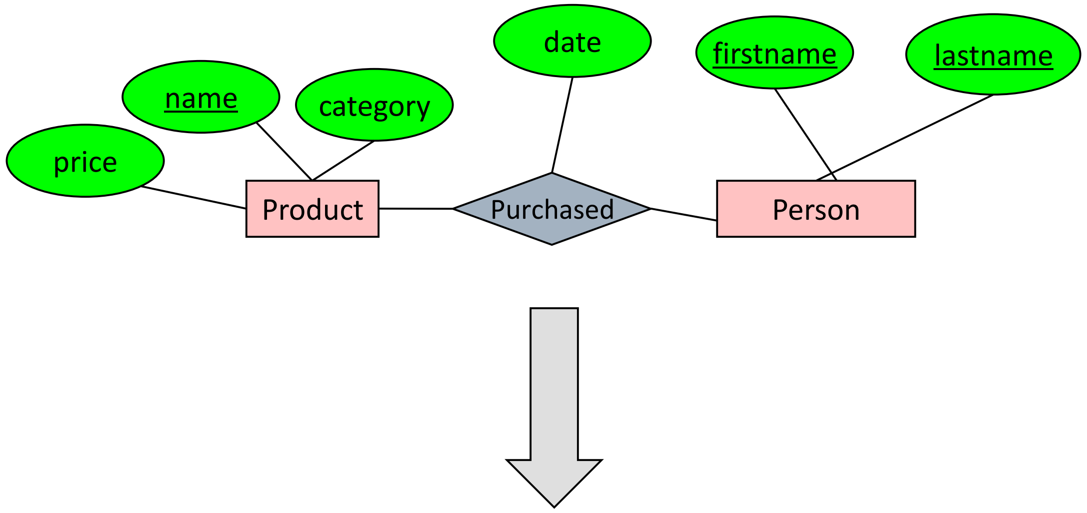


Product

<u>name</u>	price	category
Gizmo1	99.99	Camera
Gizmo2	19.99	Edible

From E/R Diagrams to Relational Schema

- A relation between entity sets $A_1, \ldots, A_N$ *also* becomes a multiset of tuples / a table
  - Each row/tuple is one relation, i.e. one unique combination of entities $(a_1,\ldots,a_N)$
  - Each row/tuple is
    - composed of the **union of the entity sets' keys**
    - has the entities' primary keys as foreign keys
    - has the union of the entity sets' keys as primary key



Purchased

name	firstname	lastname	date
Gizmo1	Bob	Alice	01/01/15
Gizmo2	Alice	Bob	01/03/15
Gizmo1	Joe	Smith	01/05/15

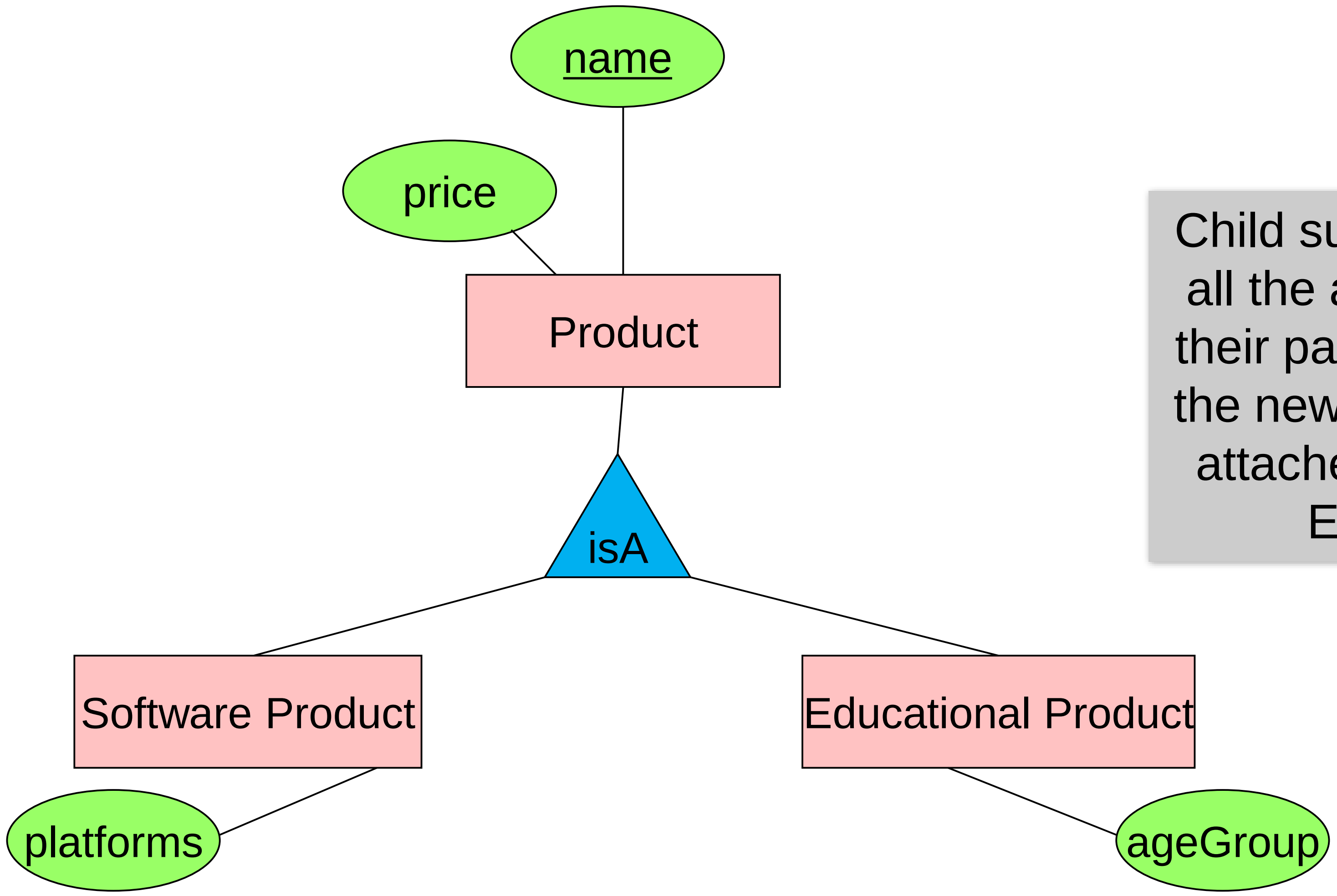
Advanced ER Concepts

1. **Subclasses**

2. **Constraints**

3. **Weak entity sets**

Modeling Subclasses



Child subclasses contain all the attributes of *all* of their parent classes **plus** the new attributes shown attached to them in the E/R diagram

Constraints in E/R Diagrams

- **Finding constraints is part of the E/R modeling process. Commonly used constraints are:**
  - Keys: Implicit constraints on uniqueness of entities
    - *Ex: An SSN uniquely identifies a person*
  - Single-value constraints:
    - *Ex: a person can have only one father*
  - Referential integrity constraints: Referenced entities must exist
    - *Ex: if you work for a company, it must exist in the database*
  - Other constraints:
    - *Ex: peoples' ages are between 0 and 150*

E/R Summary

- **E/R diagrams are a visual syntax that allows technical and non-technical people to talk**
  - For conceptual design
- **Basic constructs: entity, relationship, and attributes**
- **A good design is faithful to the constraints of the application**

Today’s Lecture

1. SQL introduction & schema definitions
2. Basic single-table queries
3. Multi-table queries

1. SQL Introduction & Definitions

What you will learn about in this section

1. What is SQL?

2. Basic schema definitions

3. Keys & constraints intro

4. ACTIVITY: CREATE TABLE statements

SQL Motivation

- **Dark times 5 years ago.**
  - Are databases dead?

- **Now, as before: everyone sells SQL**
  - Pig, Hive, Impala

- **“Not-Yet-SQL?”**

Basic SQL

SQL Introduction

- **SQL is a standard language for querying and manipulating data**

- **SQL is a very high-level programming language**
  - This works because it is optimized well!

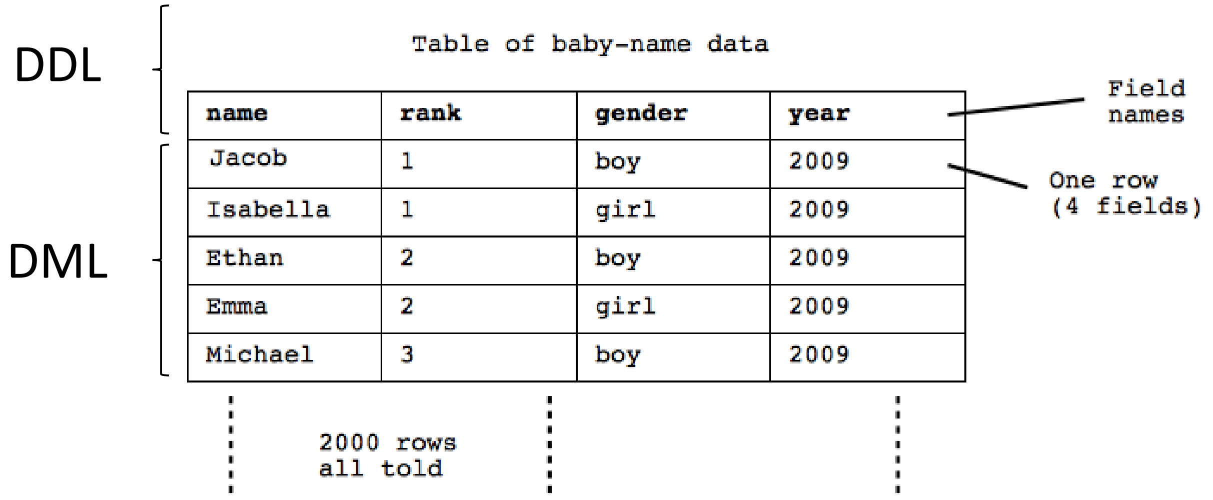
- **Many standards out there:**
  - ANSI SQL, SQL92 (a.k.a. SQL2), SQL99 (a.k.a. SQL3), ….
  - Vendors support various subsets

SQL stands for **S**tructured **Q**uery **L**anguage

SQL is a…

- **Data Definition Language (DDL)**
  - Define relational *schemata*
  - Create/alter/delete tables and their attributes

- **Data Manipulation Language (DML)**
  - Insert/delete/modify tuples in tables
  - Query one or more tables



name	rank	gender	year
Jacob	1	boy	2009
Isabella	1	girl	2009
Ethan	2	boy	2009
Emma	2	girl	2009
Michael	3	boy	2009

Tables in SQL

Product

PName	Price	Manufacturer
Gizmo	$19.99	GizmoWorks
Powergizmo	$29.99	GizmoWorks
SingleTouch	$149.99	Canon
MultiTouch	$203.99	Hitachi

A **relation** or **table** is a multiset of tuples having the attributes specified by the schema

Let’s break this definition down

Tables in SQL

Product

PName	Price	Manufacturer
Gizmo	$19.99	GizmoWorks
Powergizmo	$29.99	GizmoWorks
SingleTouch	$149.99	Canon
MultiTouch	$203.99	Hitachi

A **multiset** is an unordered list (or: a set with multiple duplicate instances allowed)

List:	[1, 1, 2, 3]
Set:	{1, 2, 3}
Multiset:	{1, 1, 2, 3}

Tables in SQL

Product

PName	Price	Manufacturer
Gizmo	$19.99	GizmoWorks
Powergizmo	$29.99	GizmoWorks
SingleTouch	$149.99	Canon
MultiTouch	$203.99	Hitachi

An **attribute** (or **column**) is a typed data entry present in each tuple in the relation

*Attributes must have an **atomic** type in standard SQL, i.e. not a list, set, etc.*

Tables in SQL

Product

PName	Price	Manufacturer
Gizmo	$19.99	GizmoWorks
Powergizmo	$29.99	GizmoWorks
SingleTouch	$149.99	Canon
MultiTouch	$203.99	Hitachi

A **tuple** or **row** is a single entry in the table having the attributes specified by the schema

Also referred to sometimes as a ***record***

Tables in SQL

Product

PName	Price	Manufacturer
Gizmo	$19.99	GizmoWorks
Powergizmo	$29.99	GizmoWorks
SingleTouch	$149.99	Canon
MultiTouch	$203.99	Hitachi

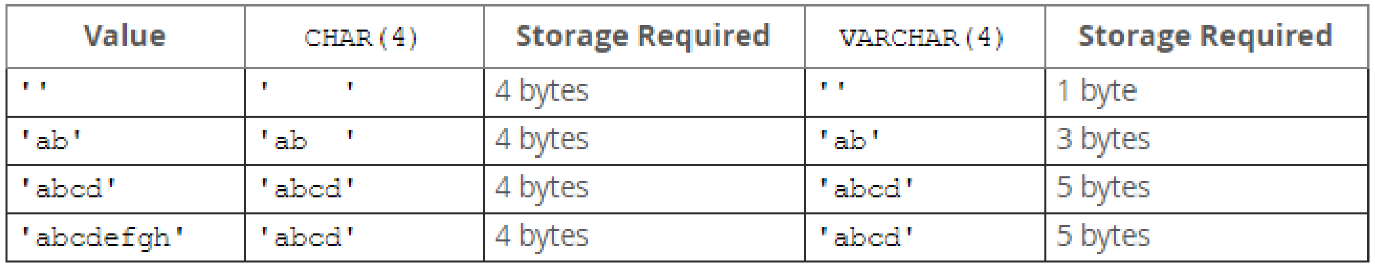
The number of tuples is the **cardinality** of the relation

The number of attributes is the **arity** of the relation

Data Types in SQL

- **Atomic types:**
  - Characters: CHAR(20), VARCHAR(50)
  - Numbers: INT, BIGINT, SMALLINT, FLOAT
  - Others: MONEY, DATETIME, …

- **Every attribute must have an atomic type**
  - Hence tables are flat



Value	`CHAR(4)`	Storage Required	`VARCHAR(4)`	Storage Required
`''`	`'    '`	4 bytes	`''`	1 byte
`'ab'`	`'ab  '`	4 bytes	`'ab'`	3 bytes
`'abcd'`	`'abcd'`	4 bytes	`'abcd'`	5 bytes
`'abcdefgh'`	`'abcd'`	4 bytes	`'abcd'`	5 bytes

Study more: https://dev.mysql.com/doc/refman/5.7/en/char.html

Table Schemas

- **The schema of a table is the table name, its attributes, and their types:**

Product(Pname: *string*, Price: *float*, Category: *string*, Manufacturer: *string*)

- **A key is an attribute whose values are unique; we underline a key**

Product(<u>Pname</u>: *string*, Price: *float*, Category: *string*, <u>Manufacturer</u>: *string*)

NULL and NOT NULL

- **To say “don’t know the value” we use NULL**
  - NULL has (sometimes painful) semantics, more detail later

Students(sid:string, name:string, gpa: float)

sid	name	gpa
123	Bob	3.9
143	Jim	NULL

Say, Jim just enrolled in his first class.

In SQL, we may constrain a column to be NOT NULL, e.g., “name” in this table

General Constraints

- **We can actually specify arbitrary assertions**
  - E.g. *“There cannot be 25 people in the DB class”*

- **In practice, we don’t specify many such constraints. Why?**
  - Performance!

Whenever we do something ugly (or avoid doing something convenient) it’s for the sake of performance

Summary of Schema Information

- **Schema and Constraints are how databases understand the semantics (meaning) of data**

- **They are also useful for optimization**

- **SQL supports general constraints:**
  - Keys and foreign keys are most important

Oracle Practice #1

Install Oracle databases

- Recommend desktop environments
- Windows version
  - https://www.oracle.com/database/technologies/oracle12c-windows-downloads.html
- Try to choose other installation options

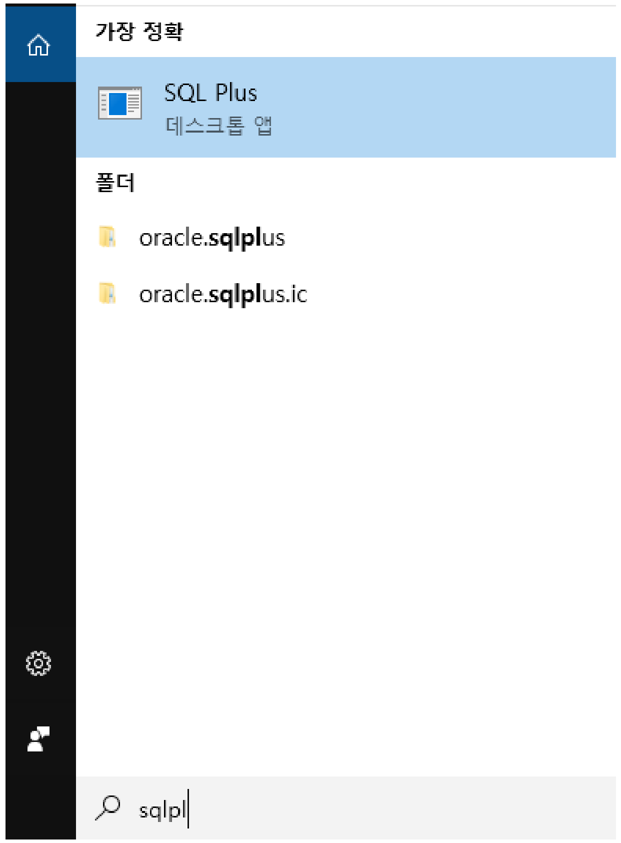
Alternatives

- Find new environments
  - E.g., laptop -> desktop, having more memories
- Virtual machines
  - If you want to use different operating systems such as Windows or Linux, e.g., from Mac, install virtual machines such as VMware or VirtualBox where you can install Windows or Linux and then install Oracle databases
- Use cloud services such as Amazon AWS, Google GCP, MS Azure with free credit
  - E.g., Amazon RDS for Oracle

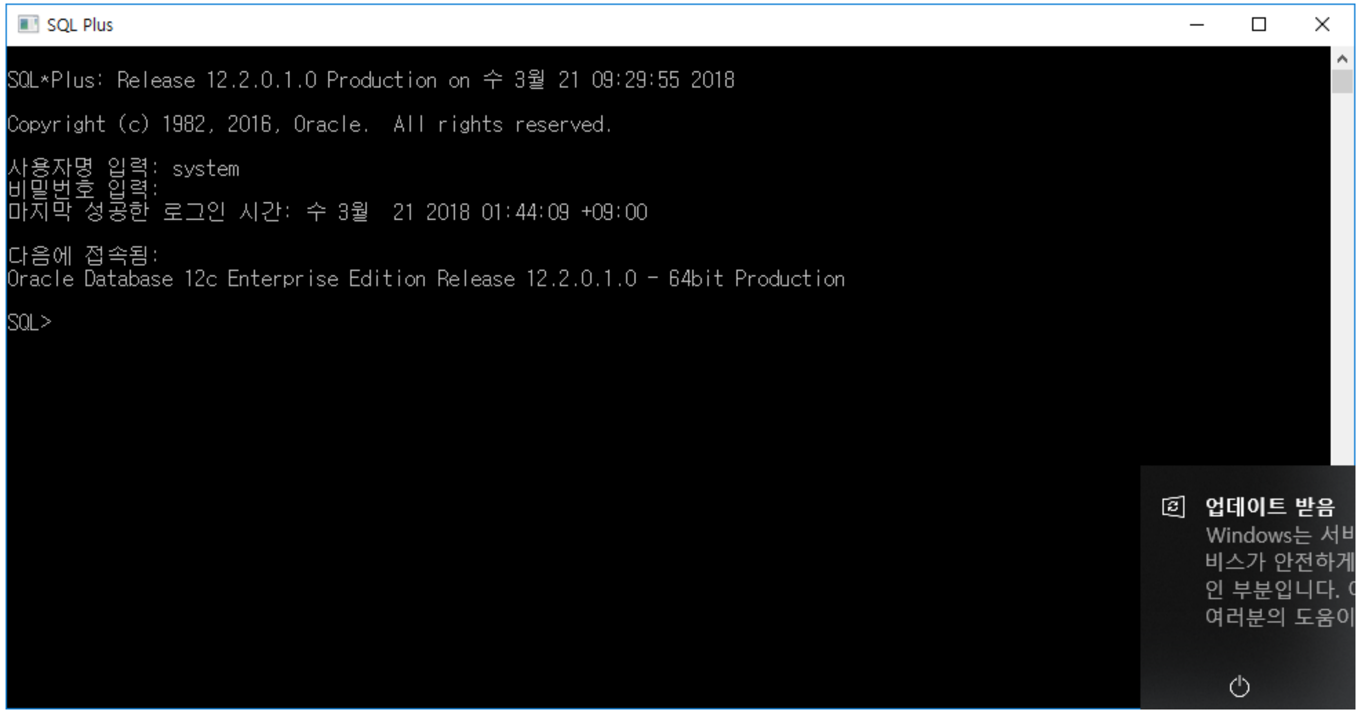
Oracle Practice #1

Start Oracle

- Execute SQL Plus



- User Authentication
  - ID: system
  - Password: oraclepractice

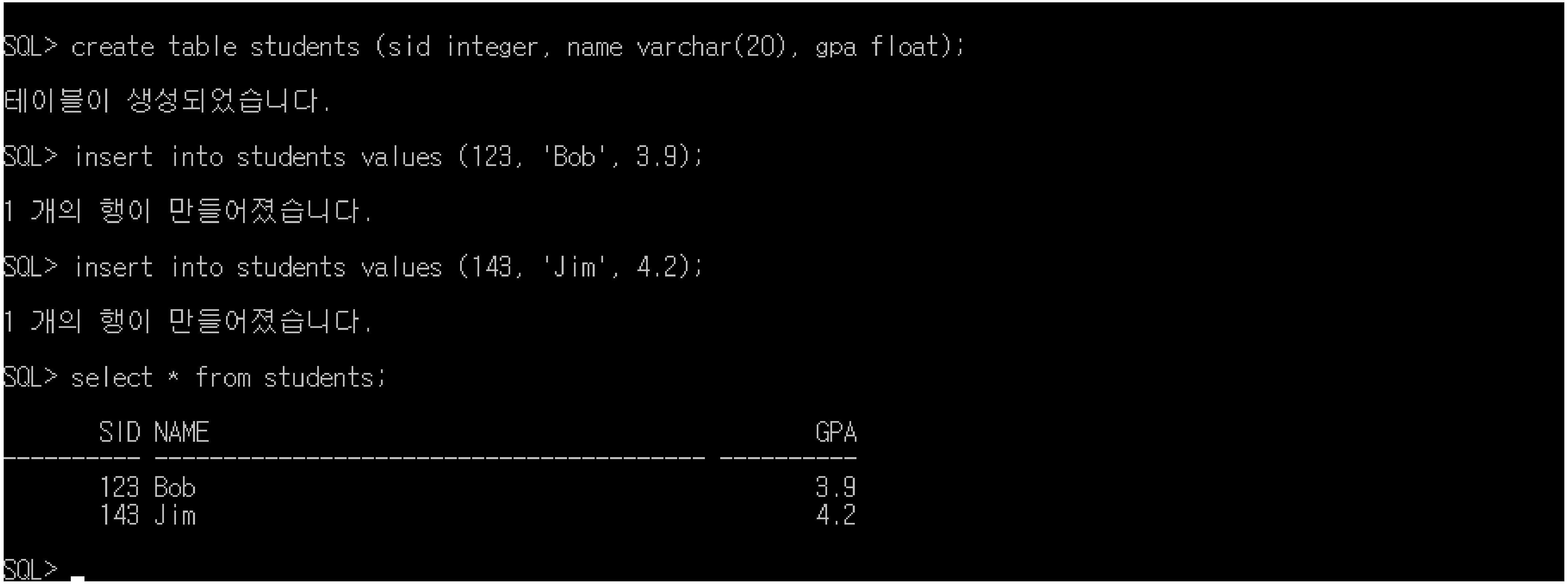


SQL Plus

```
SQL*Plus: Release 12.2.0.1.0 Production on 수 3월 21 09:29:55 2018

Copyright (c) 1982, 2016, Oracle.  All rights reserved.

사용자명 입력: system
비밀번호 입력:
마지막 성공한 로그인 시간: 수 3월  21 2018 01:44:09 +09:00

다음에 접속됨:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL>
```

Oracle Practice #1

Test Oracle



```
SQL> create table students (sid integer, name varchar(20), gpa float);

테이블이 생성되었습니다.

SQL> insert into students values (123, 'Bob', 3.9);

1 개의 행이 만들어졌습니다.

SQL> insert into students values (143, 'Jim', 4.2);

1 개의 행이 만들어졌습니다.

SQL> select * from students;

       SID NAME                                            GPA
---------- ---------------------------------------- ----------
       123 Bob                                             3.9
       143 Jim                                             4.2

SQL> _
```

Oracle Practice #1

- **Test yourself**
  - Make the following table using CREATE TABLE

Product

PName	Price	Manufacturer
Gizmo	$19.99	GizmoWorks
Powergizmo	$29.99	GizmoWorks
SingleTouch	$149.99	Canon
MultiTouch	$203.99	Hitachi

Oracle Practice #1

- **Build database with real data**
  1. Download ACDB.sql from e-class
  2. Copy ACDB.sql into a specific folder (e.g., c:/work/ACDB.sql)
  3. In SQLPlus, execute the following command
     - @c:/work/ACDB.sql
     - If some problems occur, execute the following command, and then execute the command above again
       - alter session set nls_language="AMERICAN";
  4. Check if data are stored correctly
     - select * from ACDB_SECTORS;
     - select * from ACDB_PACKAGES;
     - select * from ACDB_CUSTOMERS;