SKILLPILLS

Skill Pill: Database Manipulation

Lecture 1: Background

Guido Klingbeil

Okinawa Institute of Science and Technology
guido-klingbeil@oist.jp

June 20, 2016

OIST

Overview

1. Introduction
   - What is a database?
   - Relational databases
2. Transactions
   - ACID properties
3. Database design
   - Design process
4. Entity relationship model
   - Entities
   - Keys
   - Relationships

What is a database system?

A database is a large, integrated collection of data:

- Banking: customer accounts and their balances,
- Airlines: reservations, schedules,
- Science:
  - SBML models and simulated data,
  - Experimental data such as gene expression profiles.

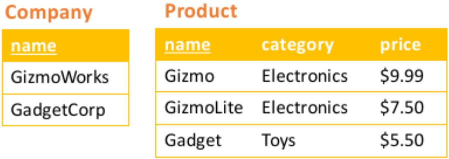
A Database Management System (DBMS) is a software designed to store, manipulate, and manage databases. Examples are Oracle, MS SQL, mySQL, or mariaDB.
We will look at relational databases.

Relational databases

We will use relational databases using the relational data model.

This model organizes data into one or more tables (or "relations") of columns and rows, with a unique key identifying each row.

Rows are also called records. Generally, each table/relation represents one "entity type" (such as company or product). The rows represent instances of that type of entity and the columns representing values attributed to that instance.



Company

name
GizmoWorks
GadgetCorp

Product

name	category	price
Gizmo	Electronics	$9.99
GizmoLite	Electronics	$7.50
Gadget	Toys	$5.50

Transactions

A transaction (TXN) is a sequence of one or more operations (read or writes) which reflects a real world transition. Two goals:

- To provide reliable units of work that allow correct recovery from failures and keep a database consistent even in cases of system failure, when execution stops (completely or partially) and many operations upon a database remain uncompleted, with unclear status.
- To provide isolation between programs accessing a database concurrently. If this isolation is not provided, the programs' outcomes are possibly erroneous.

Examples:

1. Credit or debit to or from an account,
2. Tracking the shipment of a parcel,
3. Create a database table.

This means:

In the real world, a transaction either happened completely or not at all.

Properties of transactions ACID

A transaction has to meet the ACID properties:

- Atomicity - the state shows either all effects of a TXN, or none.
- Consistency - A TXNmoves a system from one consistent state to the next.
- Isolation - The effects of multiple transactions is the same as the run one after the other.
- Durability - Ones a TXN is commited, its effect remein in the database.

Atomicity

Atomicity requires that each transaction be "all or nothing": if one part of the transaction fails, then the entire transaction fails, and the database state is left unchanged. An atomic system must guarantee atomicity in each and every situation, including power failures, errors, and crashes. To the outside world, a committed transaction appears (by its effects on the database) to be indivisible ("atomic"), and an aborted transaction does not happen.

Consistency

The consistency property ensures that any transaction will bring the database from one valid state to another. Any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof. This does not guarantee correctness of the transaction in all ways the application programmer might have wanted (that is the responsibility of application-level code) but merely that any programming errors cannot result in the violation of any defined rules.

Isolation

The isolation property ensures that the concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially, i.e., one after the other. Providing isolation is the main goal of concurrency control. Depending on the concurrency control method (i.e., if it uses strict - as opposed to relaxed - serializability), the effects of an incomplete transaction might not even be visible to another transaction.

Durability

The durability property ensures that once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors. In a relational database, for instance, once a group of SQL statements execute, the results need to be stored permanently (even if the database crashes immediately thereafter). To defend against power loss, transactions (or their effects) must be recorded in a non-volatile memory.

Ths all sounds very technical - and it is. Luckily, your database system takes care of it and we can happily move on to database design. However, you need to be aware it. For example for analysis, debugging, or optimization.

Desiging a database consists (roughly) of three steps:

1. Requirement analysis.
2. Conceptual design.
3. Logical. physical, security design (we will not cover this).

Database design - requirement analysis

1. Requirement analysis
    - What is going to be stored?
    - How is it going to be used?
    - What are we going to do with it?
    - Who should have access to the data?

Database design - conceptual design

1. Conceptual design
    - A high level description of the database.
    - Sufficiently precise that technical people can understand it.
    - Not so precise that non-technical people are unable to participate.

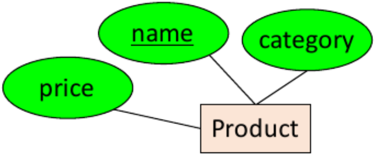
Here, the Entity relationship model, a visual syntax for database design, fits in.
"The Entity-Relationship model toward a unified view of data" Peter Chen, 1976. Cited 9830 times - one of the most cited papers in computer science.

Entities and entity sets

Entities and entity sets are the primitive unit of the E/R model. Entities are the individual objects, which are members of entity sets, such as a specific person or product.

Entity sets are the classes or types of objects in our model, such as Person, Product.

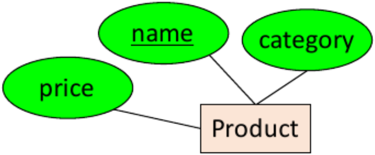
They are shown as rectangles in E/R diagrams.
Entity sets represent the sets of all possible entities.



Keys

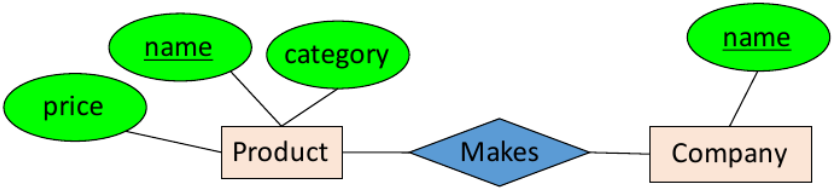
Entities can have attributes, such as price, name, category.

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

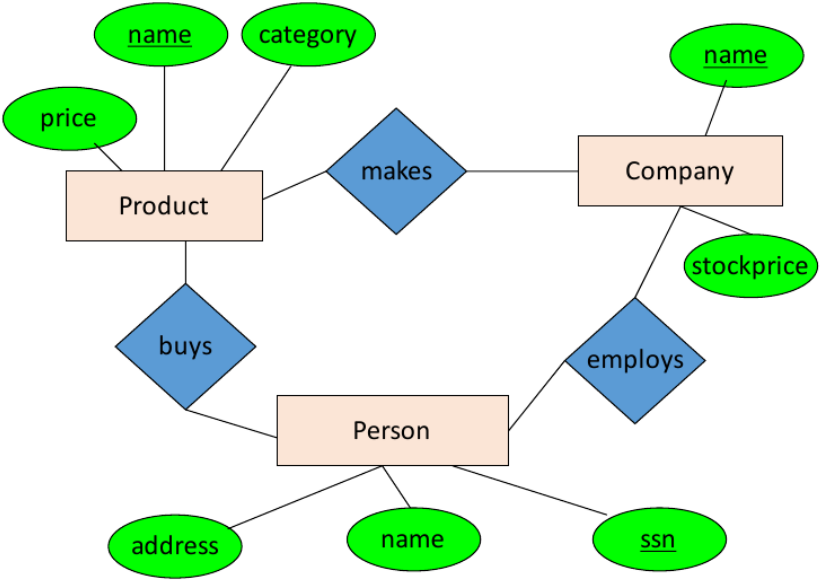


Relationships

A relationship is between two entities. Such as a product is made by a company.
A relationship between entity sets Product and Company is a subset of all possible pairs of entities in Products and Companies, with tuples uniquely identified by Product's and Companies' keys.



Relationships



Relationships

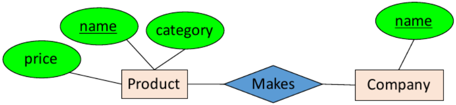
For relationship we need tables:

Company

name
GizmoWorks
GadgetCorp

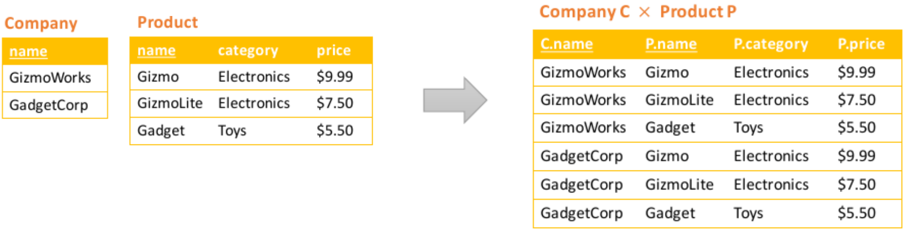
Product

name	category	price
Gizmo	Electronics	$9.99
GizmoLite	Electronics	$7.50
Gadget	Toys	$5.50



Relationships

For relationship we need tables:



Company

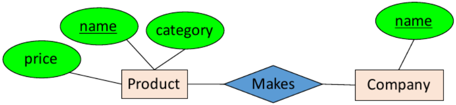
<u>name</u>
GizmoWorks
GadgetCorp

Product

<u>name</u>	category	price
Gizmo	Electronics	$9.99
GizmoLite	Electronics	$7.50
Gadget	Toys	$5.50

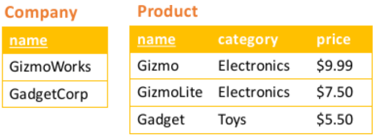
Company C × Product P

<u>C.name</u>	<u>P.name</u>	P.category	P.price
GizmoWorks	Gizmo	Electronics	$9.99
GizmoWorks	GizmoLite	Electronics	$7.50
GizmoWorks	Gadget	Toys	$5.50
GadgetCorp	Gizmo	Electronics	$9.99
GadgetCorp	GizmoLite	Electronics	$7.50
GadgetCorp	Gadget	Toys	$5.50



Relationships

For relationship we need tables:

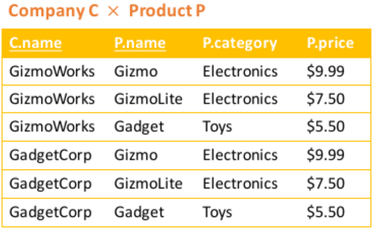


Company

<u>name</u>
GizmoWorks
GadgetCorp

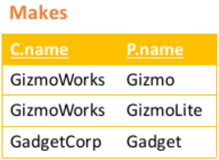
Product

<u>name</u>	category	price
Gizmo	Electronics	$9.99
GizmoLite	Electronics	$7.50
Gadget	Toys	$5.50



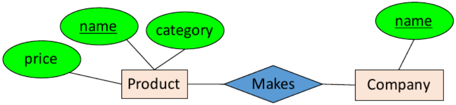
Company C × Product P

<u>C.name</u>	<u>P.name</u>	P.category	P.price
GizmoWorks	Gizmo	Electronics	$9.99
GizmoWorks	GizmoLite	Electronics	$7.50
GizmoWorks	Gadget	Toys	$5.50
GadgetCorp	Gizmo	Electronics	$9.99
GadgetCorp	GizmoLite	Electronics	$7.50
GadgetCorp	Gadget	Toys	$5.50



Makes

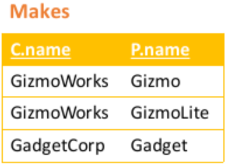
<u>C.name</u>	<u>P.name</u>
GizmoWorks	Gizmo
GizmoWorks	GizmoLite
GadgetCorp	Gadget



Relationships

What is the key?

The union of C.name and P.name is the key. In the context of relational databases, a foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table.



Makes

C.name	P.name
GizmoWorks	Gizmo
GizmoWorks	GizmoLite
GadgetCorp	Gadget

Exercise

Pick some of the first of:

Using pencil and paper pick some of these exercises and draw the ER diagram (Note: some of the later problems require some UML (Unified Modelling Language) knowledge).
`http://db4u.wikidot.com/erexercises`

For the ones you chose:

Using pencil and paper and design the table strcuture.
`http://db4u.wikidot.com/erexercises`