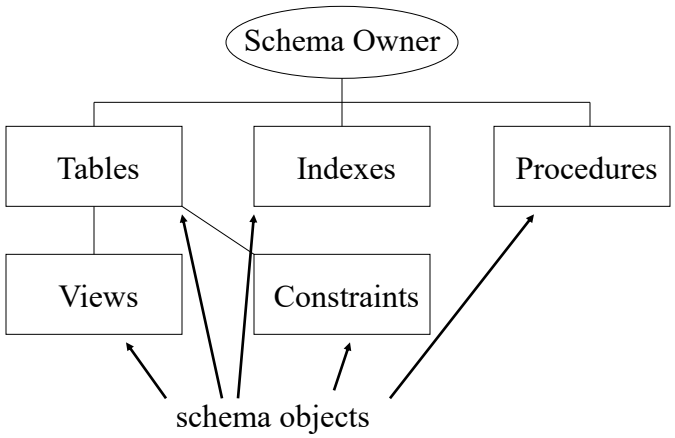
Databases & SQL

1

A Database Schema



[1]

1) Stephens, R.K. and Plew. R.R., 2001. *Database Design*. SAMS, Indianapolis , IN. (with slight changes by V.G.D.)

2

Table

- "A *table* is the primary unit of physical storage for data in a database."[1]
- Usually a database contains more than one table.

1) Stephens, R.K. and Plew. R.R., 2001. *Database Design*. SAMS, Indianapolis , IN.

3

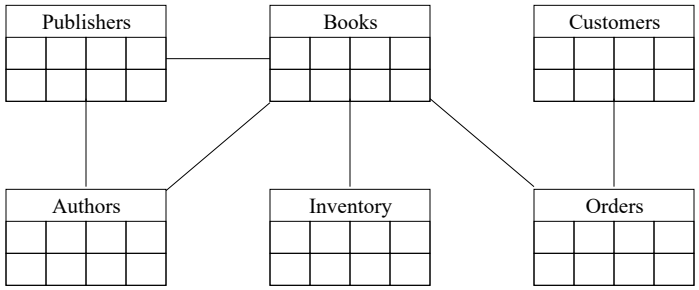
Table

Name	Company	Phone Number	E-mail Address
Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com

4

A Database with Multiple Tables

[1]



1) Stephens, R.K. and Plew. R.R., 2001. *Database Design*. SAMS, Indianapolis , IN. (with slight changes by V.G.D.)

5

Table

Customers

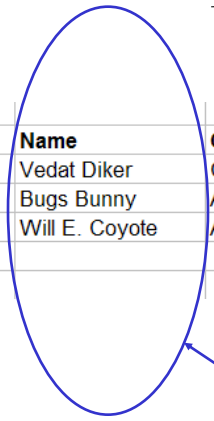
Name	Company	Phone Number	E-mail Address
Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com

6

Field (Column)

Customers

Name	Company	Phone Number	E-mail Address
Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com



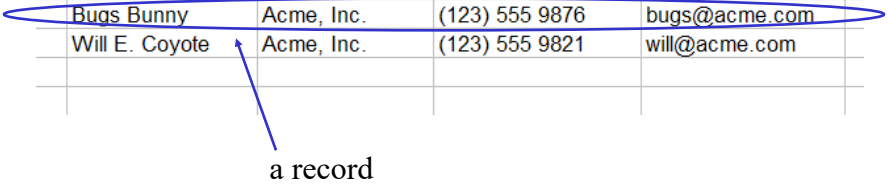
a field

7

Record (Row)

Customers

Name	Company	Phone Number	E-mail Address
Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com



a record

8

Primary Key

Customers

Customer ID	Name	Company	Phone Number	E-mail Address
6273	Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
3245	Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
1324	Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com

primary key field

Primary key is a unique identifier of records in a table.

Primary key values may be generated manually or automatically.

9

Primary Key

Roles (Performances)

Actor/Actress	Movie	Character Name
Keanu Reeves	Matrix	Neo
Laurence Fishburne	Matrix	Morpheus
Carrie-Anne Moss	Matrix	Trinity
Keanu Reeves	Sweet November	Nelson Moss
Charlize Theron	Sweet November	Sara Deever
Charlize Theron	Waking Up in Reno	Candy Kirkendall
Laurence Fishburne	Othello	Othello
Ted Lange	Othello	Othello

primary key fields

A primary key can consist of more than one field.

10

SQL Introduction

Standard language for querying and manipulating data

Structured **Q**uery **L**anguage

Many standards out there:
- ANSI SQL
- SQL92 (a.k.a. SQL2)
- SQL99 (a.k.a. SQL3)
- Vendors support various subsets of these
- What we discuss is common to all of them

11

11

SQL

- Data Definition Language (DDL)
  - Create/alter/delete tables and their attributes
  - Following lectures...
- Data Manipulation Language (DML)
  - Query one or more tables – discussed next !
  - Insert/delete/modify tuples in tables
- Transact-SQL
  - Idea: package a sequence of SQL statements → server
  - Won't discuss in class

12

12

Data in SQL

1. Atomic types, a.k.a. data types
2. Tables built from atomic types

Unlike XML, no nested tables, only flat tables are allowed!

- We will see later how to decompose complex structures into multiple flat tables

13

13

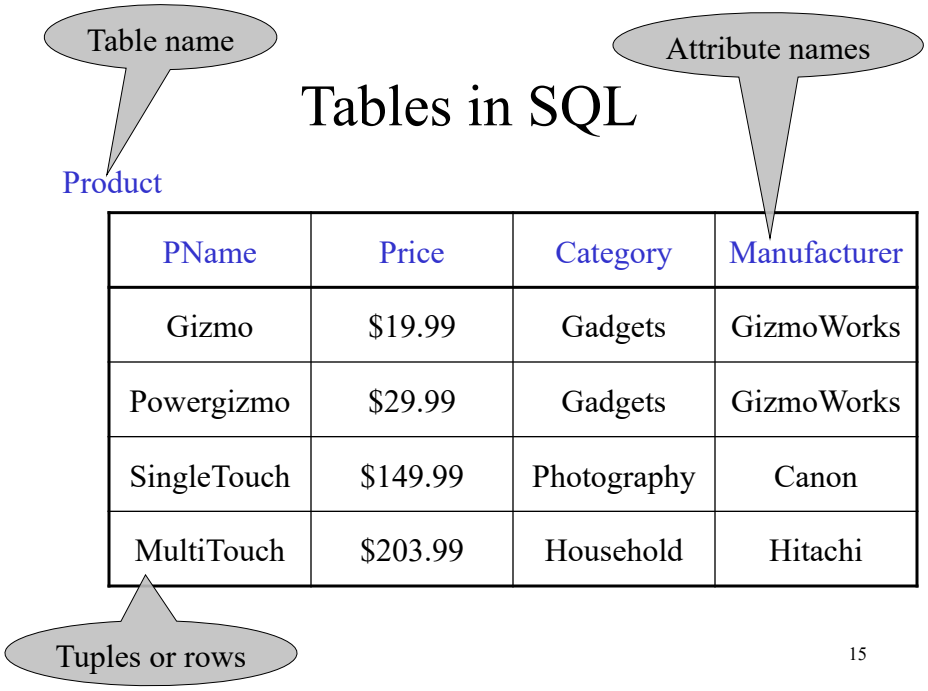
Data Types in SQL

- Characters:
  - CHAR(20) -- fixed length
  - VARCHAR(40) -- variable length
- Numbers:
  - BIGINT, INT, SMALLINT, TINYINT
  - REAL, FLOAT -- differ in precision
  - MONEY
- Times and dates:
  - DATE
  - DATETIME -- SQL Server
- Others... All are simple

14

14

Tables in SQL



15

15

Tables Explained

- A tuple = a record
  - Restriction: all attributes are of atomic type
- A table = a set of tuples
  - Like a list…
  - …but it is unordered: no **first()**, no **next()**, no **last()**.

16

16

Tables Explained

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

Product(PName, Price, Category, Manfacturer)

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

Product(PName, Price, Category, Manfacturer)

17

17

SQL Query

Basic form: (plus many many more bells and whistles)

```
SELECT  attributes
FROM    relations (possibly multiple)
WHERE   conditions (selections)
```

18

18

Simple SQL Query

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

```
SELECT  *
FROM    Product
WHERE   category='Gadgets'
```

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks

"selection"

19

19

Simple SQL Query

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

```
SELECT  PName, Price, Manufacturer
FROM    Product
WHERE   Price > 100
```

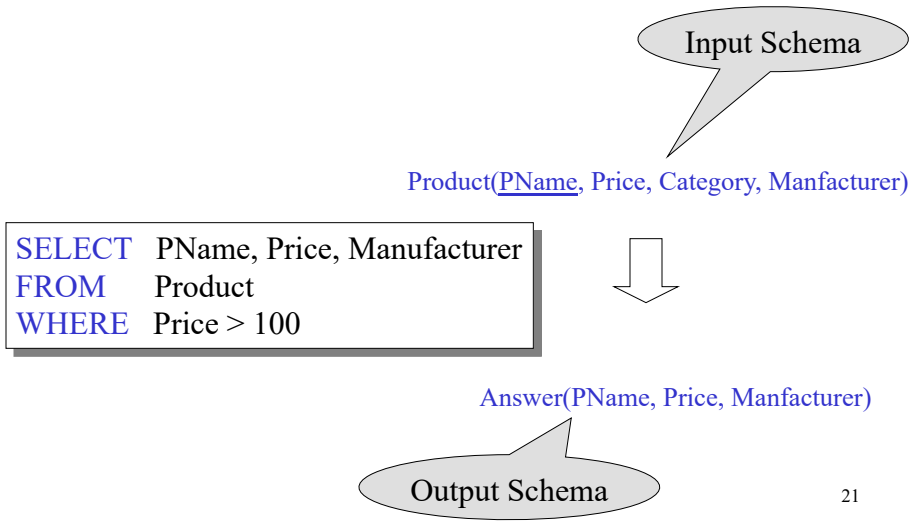
PName	Price	Manufacturer
SingleTouch	$149.99	Canon
MultiTouch	$203.99	Hitachi

"selection" and "projection"

20

20

A Notation for SQL Queries



21

21

Selections

What goes in the WHERE clause:

- x = y, x < y, x <= y, etc
  - For number, they have the usual meanings
  - For CHAR and VARCHAR: lexicographic ordering
    - Expected conversion between CHAR and VARCHAR
  - For dates and times, what you expect...
- Pattern matching on strings...

22

22

The **LIKE** operator

- s **LIKE** p: pattern matching on strings
- p may contain two special symbols:
  - % = any sequence of characters
  - _ = any single character

Product(PName, Price, Category, Manufacturer)
Find all products whose name mentions 'gizmo':

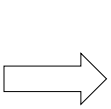
```
SELECT  *
FROM    Products
WHERE   PName LIKE '%gizmo%'
```

23

23

Eliminating Duplicates

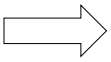
```
SELECT  DISTINCT category
FROM    Product
```



Category
Gadgets
Photography
Household

Compare to:

```
SELECT  category
FROM    Product
```



Category
Gadgets
Gadgets
Photography
Household

24

24

Ordering the Results

```
SELECT   pname, price, manufacturer
FROM     Product
WHERE    category='gizmo' AND price > 50
ORDER BY price, pname
```

Ordering is ascending, unless you specify the DESC keyword.

Ties are broken by the second attribute on the ORDER BY list, etc.

25

25

Joins in SQL

- Connect two or more tables:

Product

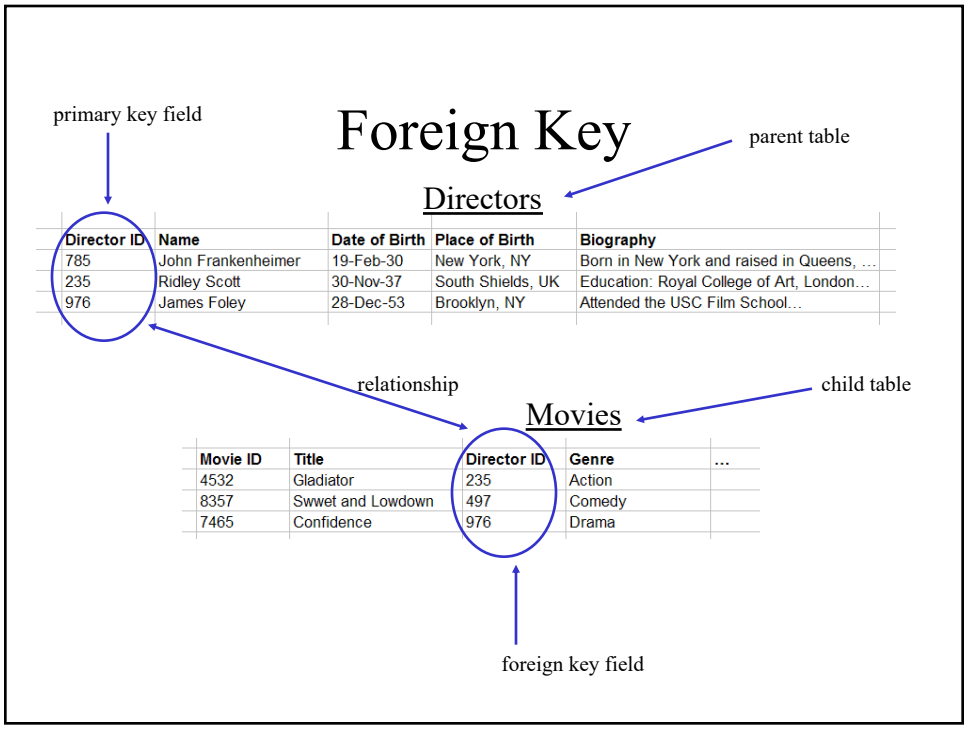
PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

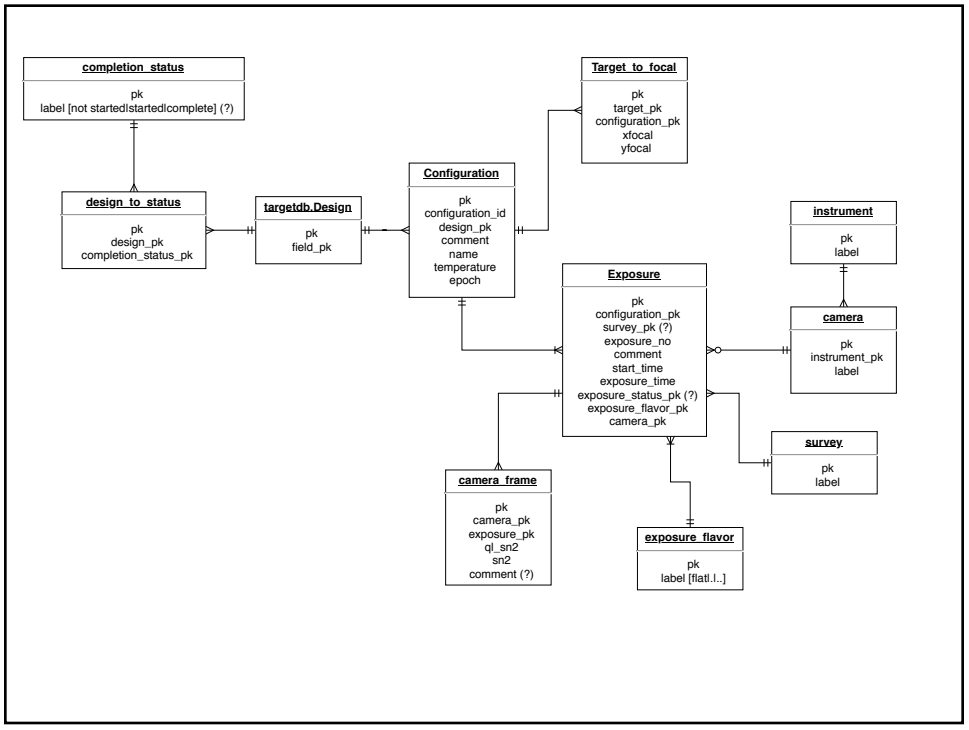
Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

What is the connection between them ?

26


primary key field
Foreign Key
parent table
Directors
Director ID | Name | Date of Birth | Place of Birth | Biography
785 | John Frankenheimer | 19-Feb-30 | New York, NY | Born in New York and raised in Queens, …
235 | Ridley Scott | 30-Nov-37 | South Shields, UK | Education: Royal College of Art, London…
976 | James Foley | 28-Dec-53 | Brooklyn, NY | Attended the USC Film School…
relationship
child table
Movies
Movie ID | Title | Director ID | Genre | …
4532 | Gladiator | 235 | Action
8357 | Swwet and Lowdown | 497 | Comedy
7465 | Confidence | 976 | Drama
foreign key field

27


completion_status
pk
label [not started|started|complete] (?)
Target_to_focal
pk
target_pk
configuration_pk
xfocal
yfocal
design_to_status
pk
design_pk
completion_status_pk
targetdb.Design
pk
field_pk
Configuration
pk
configuration_id
design_pk
comment
name
temperature
epoch
instrument
pk
label
Exposure
pk
configuration_pk
survey_pk (?)
exposure_no
comment
start_time
exposure_time
exposure_status_pk (?)
exposure_flavor_pk
camera_pk
camera
pk
instrument_pk
label
survey
pk
label
camera_frame
pk
camera_pk
exposure_pk
ql_sn2
sn2
comment (?)
exposure_flavor
pk
label [flat|..]

28

Joins

Product (pname, price, category, manufacturer)
Company (cname, stockPrice, country)

Find all products under $200 manufactured in Japan; return their names and prices.

Join between Product and Company

```
SELECT   pname, price
FROM     Product, Company
WHERE    manufacturer=cname AND country='Japan'
         AND price <= 200
```

29

29

Joins in SQL

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT   pname, price
FROM     Product, Company
WHERE    manufacturer=cname AND country='Japan'
         AND price <= 200
```

PName	Price
SingleTouch	$149.99

30

30

Joins

Product (pname, price, category, manufacturer)
Company (cname, stockPrice, country)

Find all countries that manufacture some product in the 'Gadgets' category.

```
SELECT   country
FROM     Product, Company
WHERE    manufacturer=cname AND category='Gadgets'
```

31

31

Joins in SQL

Product

Name	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT   country
FROM     Product, Company
WHERE    manufacturer=cname AND category='Gadgets'
```

What is the problem ? What's the solution ?

Country
??
??

32

32

Joins

Product (pname, price, category, manufacturer)
Purchase (buyer, seller, store, product)
Person(persname, phoneNumber, city)

Find names of people living in Seattle that bought some product in the 'Gadgets' category, and the names of the stores they bought such product from

```
SELECT   DISTINCT persname, store
FROM     Person, Purchase, Product
WHERE    persname=buyer AND product = pname AND
         city='Seattle'  AND category='Gadgets'
```

33

When are two tables related?

- You guess they are
- I tell you so
- Foreign keys are a method for schema designers to tell you so
  - A foreign key states that a column is a reference to the key of another table
    ex: Product.manufacturer is foreign key of Company
  - Gives information and enforces constraint

34

34