Big Data Technology and its Applications

Introduction to Database

张宁 ningzhang@tsinghua.edu.cn

Outline

- Overview
- Relational data model
- Relational operator
- SQL
- Sqlite3 in python

About Database

- Database (DB): a collection of information in a structured way.

  A shared collection of logically related data, designing to meet the information needs of an organization.

- Database Management System (DBMS)
  - A software system that enables users to define, create, and maintain the database and provides controlled access to this database.
  - Typical DBMS: DB2, Oracle, MS SQL Server, MySQL, MS Access
  - Dead DBMS: Sybase, Informix
- Important functions of DBMS:
  - Data storage, retrieval, and update (create, insert, read, update, delete)
  - Transaction support

    Ensure that either all the updates corresponding to a given transaction are made or that none of them are made.

  - Concurrency control services

About Database

- Data collection: multi-source heterogeneous data
  - Heterogeneous:
    - Structured (e.g., table)
    - Semi-structured (e.g., xml file)
    - Unstructured (e.g., report, image)
  - Multi-source:
    - Business report (structured)
    - Questionnaire (structured, unstructured)
    - Internet (semi-structured, unstructured)
    - ……

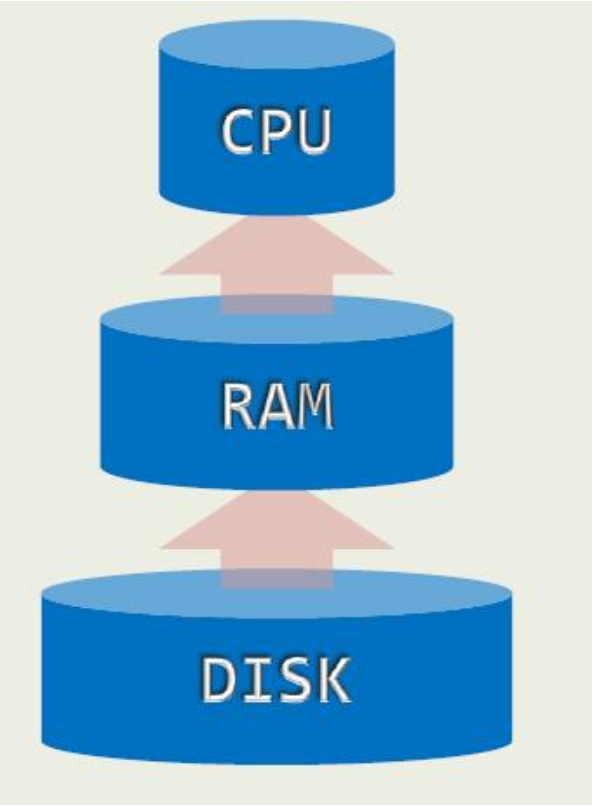
- Data collection -> Data cleaning -> Data Integration -> Database

What is the difference?

- Capacity (data / ram)
  - 1980: 10 MB / 1 MB = 10
  - 2010: 1TB / 1 GB = 1000

- RAM v.s. Disk

  ms / ns > 10^5



Outline

- Overview
- Relational data model
- Relational operator
- SQL
- Sqlite3 in python

Field and Cartesian product

- Field: a set of values with the same data type
  - $D \coloneqq \{d: d \text{ is the student's name in this classroom}\}$ is a field.
- Cartesian product
  - Given fields $D_1, D_2, \dots, D_n$, the Cartesian product of the n fields is
    $$D_1 \times D_2 \times \cdots \times D_n \coloneqq \{(d_1, d_2, \dots, d_n): d_i \in D_i, i = 1,2, \dots, n\}.$$
  - Each element $(d_1, d_2, \dots, d_n)$ is called a tuple; d_i is called a component.
- Given two fields
  - $D_1 = \{\text{Abel, Bob, Carrol}\}$,
  - $D_2 = \{\text{Power system analysis, Big data technology}\}$
- the Cartesian product is
  - $D_1 \times D_2$ = {(Abel, Power system analysis), (Abel, Big data technology), (Bob, Power system analysis), (Bob, Big data technology), (Carrol, Power system analysis), (Carrol, Big data technology)}.

Relational data model

- A relation $r(D_1, D_2, \ldots, D_n)$ represents the relationship among a subset of $D_1 \times D_2 \ldots \times D_n$.
- Loosely speaking, a relation is equivalent to a table.
- Each instance $(d_1, d_2, \ldots, d_n)$ of $r(D_1, D_2, \ldots, D_n)$ represents a row in the table.
- Each field D_i represents a column of the table.
- The column name is called attribute.

Relational data model

- Superkey
  - A superkey is a set of attributes that uniquely identifies each tuple of a relation.
- Candidate key (or key for short)
  - A candidate key is a minimal superkey.
- Primary key
  - A primary key is a specific choice of candidate keys.
- Foreign key
  - A foreign key is a set of attributes in a table that refers to the primary key of another table.

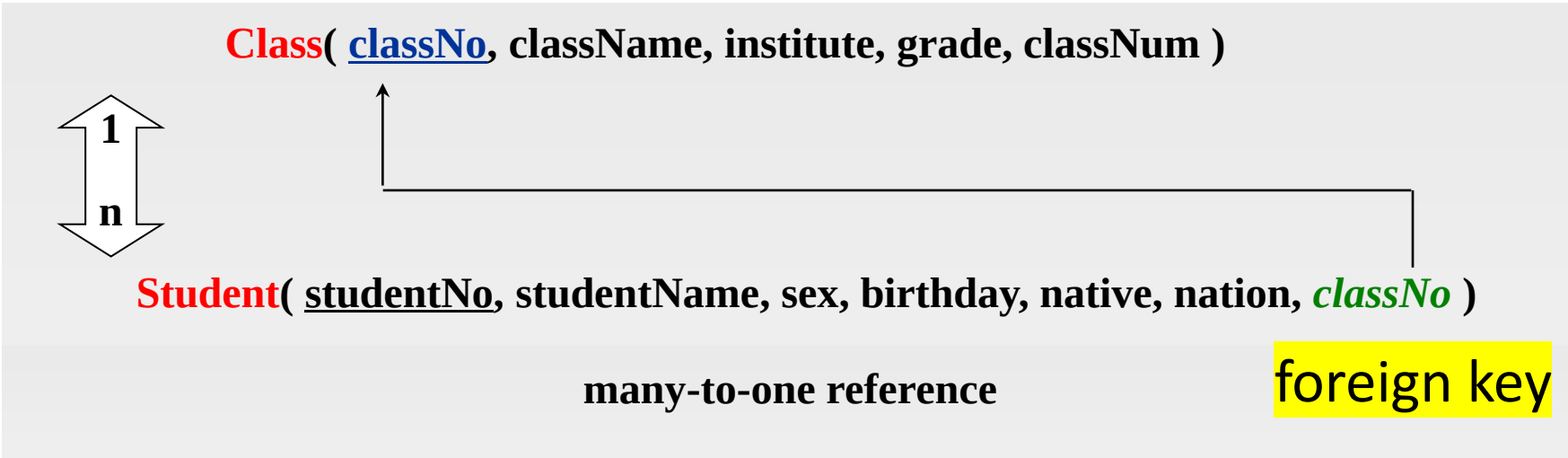
Relational data model

Student (studentNo, studentName, sex, birthday, native, nation, classNo)

- Superkey
  (studentNo), (studentNo, studentName), (studentNo, studentName, sex), …
- Candidate key
  (studentNo)
- Primary key
  (studentNo)

Relational data model

primary key

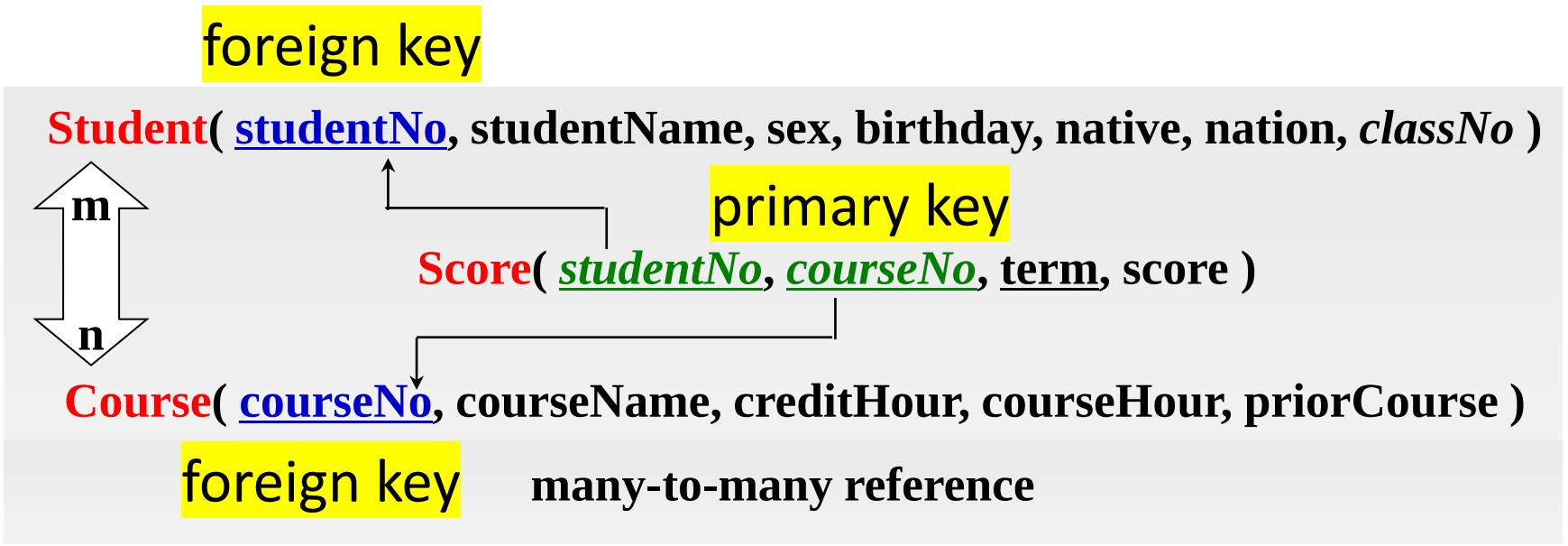


Referenced relation

Reference relation

foreign key

Relational data model

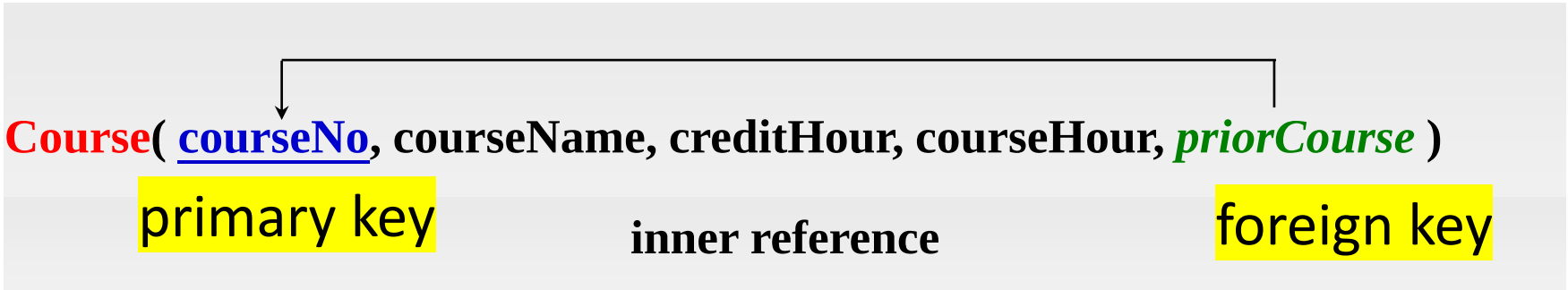


Reference relation

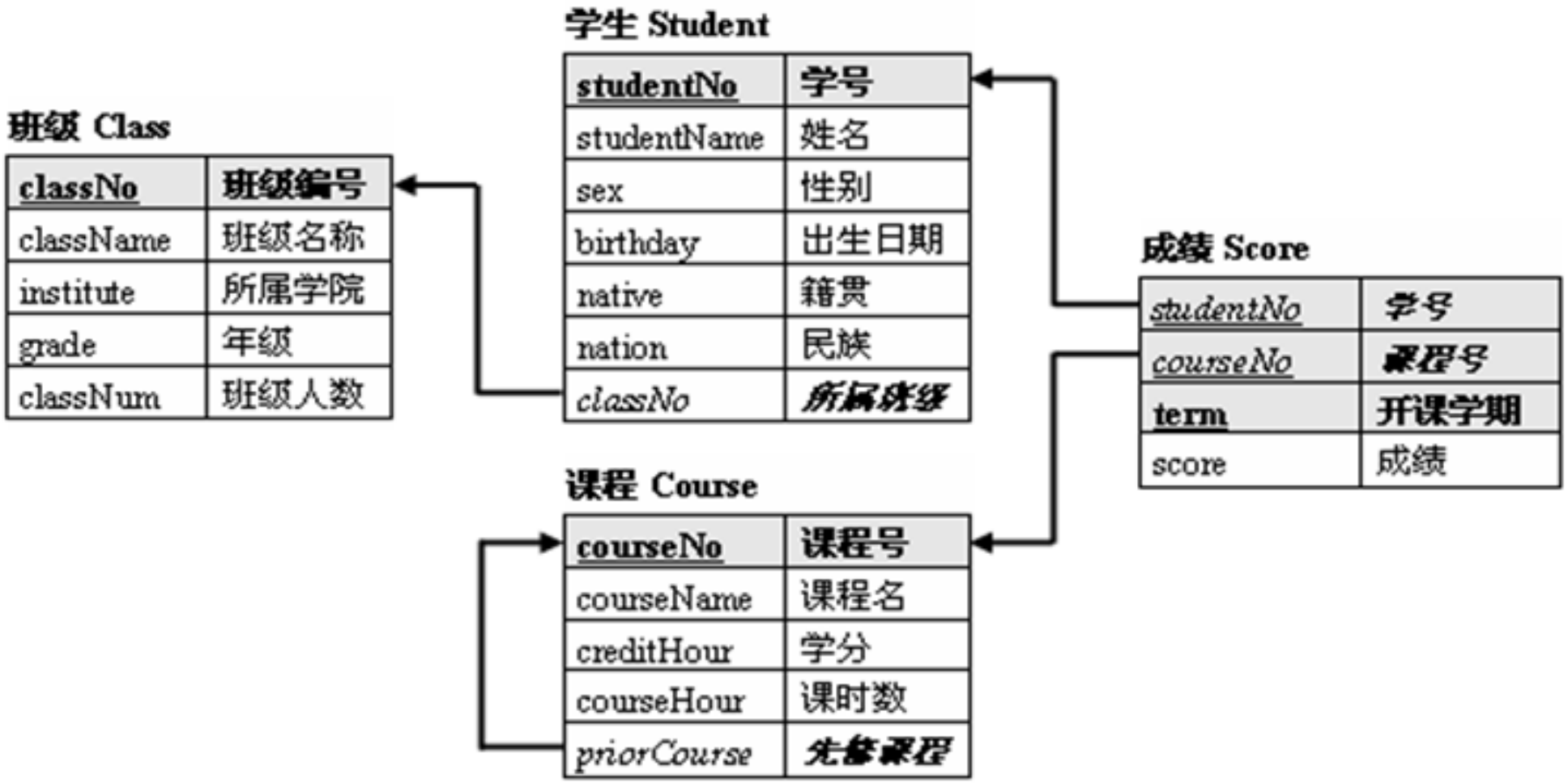
Referenced relation

Reference relation

Relational data model



Relational data model



成绩管理数据库 ScoreDB 的模式导航图

Outline

- Overview
- Relational data model
- Relational operator
- SQL
- Sqlite3 in python

Relational operators

Operators in Relational Algebra					
type	symbol	name	type	symbol	name
Set operator	$\cup$	union	Comparison operator	$>$	greater than
	$-$	minus		$\geqslant$	no less than
	$\cap$	intersect		$<$	less than
	$\times$	Cartesian product		$\leqslant$	no greater than
Relational operator	σ	select		$=$	equal
	$\prod$	project		$\neq$	not equal
	$\bowtie$	join	Logical operator	$\neg$	not
	$\div$	divide		$\wedge$	and
				$\vee$	or

Relational operators

- Select σ: for a relation r, select all instances t satisfying predicate P

$$\sigma_P(r) = \{t: t \in r \ \wedge \ P(t)\}$$

- Project $\prod$: select a subset A of the fields of relation r to form a new relation

$$\prod_A(r) = \{t[A]: t \in r\}$$

- Join $\bowtie$: join two relations r, s according to predicate P

$$r \bowtie_P s = \sigma_P(r \times s)$$

- Image Im: for a relation $r(R)$, $S \subset R$ is a subset of the fields R, the image of $x \in S$ on relation r is defined as

$$\mathrm{Im}(x|r) = \{t[R - S]: t \in r, t[S] = x\}$$

- Divide $\div$: suppose two relations $r(R), s(S)$ and that $S \subset R$

$$r \div s = \{t[R - S]: t \in r \wedge \mathrm{Im}(t[R - S] \mid r) \supset s\}$$

Relational operators

Class relation

ClassNo	ClassName	institute	grade	ClassNum
AC0703	会计学08(3)班	会计学院	2007	46
CS0701	计算机07(1)班	信息学院	2007	48
ISO802	信息系统08(2)班	信息学院	2008	43

$\sigma_{\text{grade}=2007}(\text{Class})$

$\prod_{\text{institude,grade}}(\text{Class})$

Relational operators

Student $\bowtie_{\text{Student.classNo=Class.classNo}}$ Class

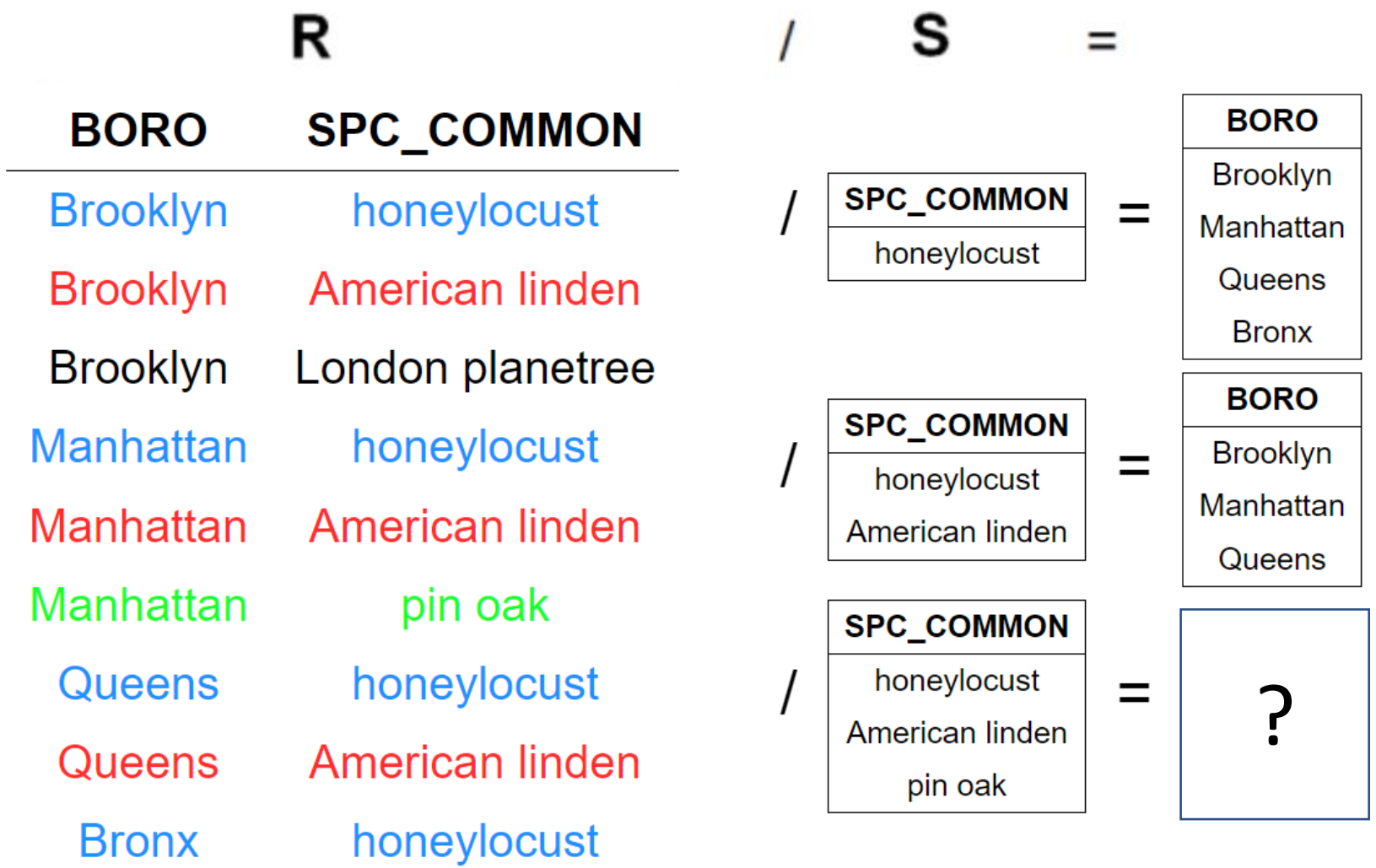
Class relation

ClassNo	ClassName	institute	grade	ClassNum

studentNo	studentName	sex	birthday	native	nation	classNo	className	institute	grade	classNum
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701	计算机07(1)班	信息学院	2007	48
701008	王红	男	1992-04-26	上海	汉族	CS0701	计算机07(1)班	信息学院	2007	48
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703	会计学08(3)班	会计学院	2007	46
703045	王红	男	1992-04-26	北京	汉族	AC0703	会计学08(3)班	会计学院	2007	46
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802	信息系统08(2)班	信息学院	2008	43
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802	信息系统08(2)班	信息学院	2008	43

0703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
0703045	王 红	男	1992-04-26	北京	汉族	AC0703
0802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
0802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

Relational operators



An example

Class关系

ClassNo	ClassName	institute	grade	ClassNum
AC0703	会计学08(3)班	会计学院	2007	46
CS0701	计算机07(1)班	信息学院	2007	48
ISO802	信息系统08(2)班	信息学院	2008	43

Student关系

StudentNo	StudentName	sex	birthday	native	nation	classNo
0701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
0701008	王 红	男	1992-04-26	上海	汉族	CS0701
0703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
0703045	王 红	男	1992-04-26	北京	汉族	AC0703
0802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
0802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

Course关系

CourseNo	CourseName	creditHour	courseHour	priorCourse
AC001	基础会计	48	3	null
CN028	大学语文	48	3	null
CS012	操作系统	80	5	null
CS015	数据库系统	64	4	CS012

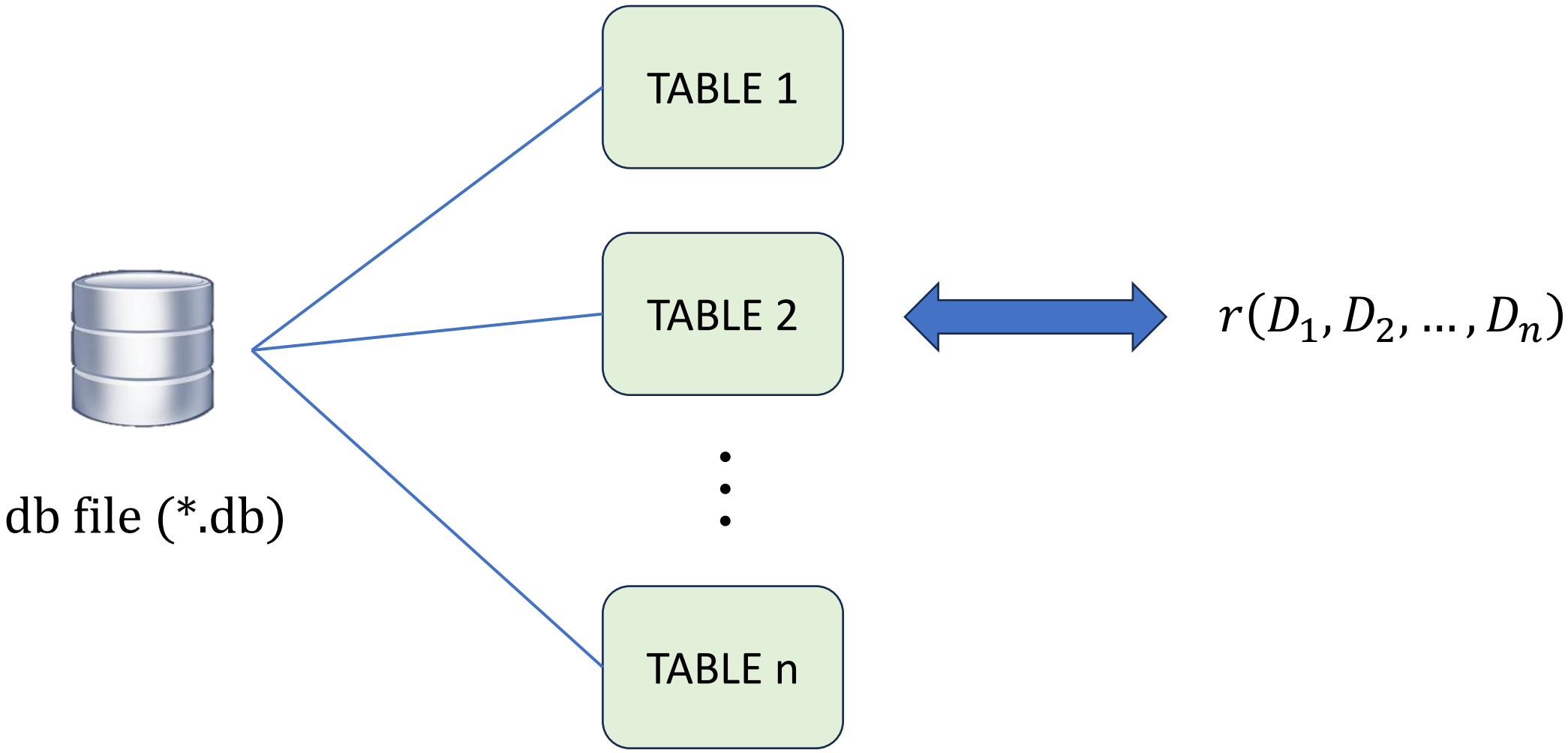
Score关系

StudentNo	courseNo	term	score
0701001	CN028	07081	85
0701001	CS012	07082	88
0701001	CS015	08091	92
0701008	AC001	07081	76
0701008	CN028	07081	86
0701008	CS012	07082	93
0701008	CS015	08091	96
0703010	AC001	07081	92
0703010	CN028	07081	83
0703010	CS012	07082	73
0703045	AC001	07081	52
0703045	AC001	08091	94
0703045	CN028	07081	80
0703045	CS015	08091	82
0802002	AC001	08091	98
0802002	CN028	08091	72
0802002	CS015	09101	85
0802005	AC001	09101	88
0802005	CS012	08092	90
0802005	CS015	09101	87

Outline

- Overview
- Relational data model
- Relational operator
- SQL
- Sqlite3 in python

Basic SQL



Basic SQL

- SELECT $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ FROM $r(D_1, D_2, \ldots, D_n)$

  Projection on $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$
- SELECT $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ FROM $r(D_1, D_2, \ldots, D_n)$ WHERE $D_{i_1} = d$

  Projection on $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ after $D_{i_1} = d$ selection
- SELECT $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ FROM $r(D_1, D_2, \ldots, D_n)$ WHERE $D_{i_1} != d$

  Projection on $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ after $D_{i_1} != d$ selection
- SELECT $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ FROM $r(D_1, D_2, \ldots, D_n)$ WHERE $D_{i_1} > d$

  Projection on $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ after $D_{i_1} > d$ selection
- SELECT $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ FROM $r(D_1, D_2, \ldots, D_n)$ ORDER BY D_{i_l} DESC, D_{i_j}

  Dictionary sorting. D_{i_l} first in descending order, D_{i_j} second in ascending order

Basic SQL

- SELECT $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ FROM $r(D_1, D_2, \ldots, D_n)$ WHERE D_{i_1} BETWEEN d AND q

  Projection on $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ after $D_{i_1} \in [d, q]$ selection
- SELECT $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ FROM $r(D_1, D_2, \ldots, D_n)$ WHERE D_{i_1} IN (d, q)

  Projection on $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ after $D_{i_1} \in \{d, q\}$ selection
- SELECT $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ FROM $r(D_1, D_2, \ldots, D_n)$ WHERE D_{i_1} NOT IN (d, q)

  Projection on $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ after $D_{i_1} \notin \{d, q\}$ selection

- SELECT $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ FROM $r(D_1, D_2, \ldots, D_n)$ WHERE D_{i_1} LIKE wildcard
- SELECT $D_{i_1}, D_{i_2}, \ldots, D_{i_k}$ FROM $r(D_1, D_2, \ldots, D_n)$ WHERE D_{i_1} REGEXP regular expression

Wildcard

- %
  - Matches any number of any characters
  - Cannot match ‘NULL’
  - Can match No character
- _
  - Matches any character
  - Can only match one character

- LIKE
  - Must match all characters of the column

Regular expression

- . (Similar to wildcard _)
- {n} (3{2} matches 33)
- {n,} (3{2,} matches 33 or 333 or 3333 or 33333 …)
- {n, m} (3{2,4} matches 33 or 333 or 3333)
- ? (is equivalent to {0, 1})
- + (is equivalent to {1,})
- * (any times, 0~∞)
- | (or)
- [3-5] (3|4|5)
- [^23] (NOT [2|3])

提示：

部分数据库（如sqlite）需要配置后才能正确使用正则表达式

https://www.sqlite.org/appfunc.html

- REGEXP
  - Matches a part of characters of the column

Outline

- Overview
- Relational data model
- Relational operator
- SQL
- Sqlite3 in python

SQLite

- What is SQLite ?

  - SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine.
  - SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.
  - SQLite source code is in the public-domain and is free to everyone to use for any purpose.

sqlite3 in python

- What is sqlite3?
  - Sqlite3 is a python module written by Gerhard Häring.
  - It provides an SQL interface compliant with the DB-API 2.0 specification described by PEP 249, and requires SQLite 3.7.15 or newer.

- Why do we use sqlite3?
  - No separate installation required.
  - High compatibility with python.
  - Lightweight.

sqlite3 in python

Import sqlite3 module

In [ ]:
```
import sqlite3
```

Connect to a database

Connect to an existing database or create a new one

In [ ]:
```
conn = sqlite3.connect('./db/demo_lecture3.db')
```

Get a cursor object to handle SQL statements

In [ ]:
```
cur = conn.cursor()
```

- If the database does not exist, sqlite3 will automatically create one.

sqlite3 in python

Get all tables in the database

In [ ]:
```
tablename = cur.execute("select name from sqlite_master where type='table' and name!='sqlite_sequence'").fetchall()
print(tablename)
```

```
[('class',), ('student',), ('course',), ('score',)]
```

Drop all existing tables

In [ ]:
```
for name in tablename:
    cur.execute(f"drop table if exists {name[0]}")
```

Load data from an excel file

In [ ]:
```
import pandas as pd

df_class = pd.read_excel('./db/demo_lecture3.xlsx', sheet_name='class')
df_student = pd.read_excel('./db/demo_lecture3.xlsx', sheet_name='student')
df_student['birthday'] = df_student['birthday'].astype(str)
df_course = pd.read_excel('./db/demo_lecture3.xlsx', sheet_name='course')
df_score = pd.read_excel('./db/demo_lecture3.xlsx', sheet_name='score')
```

Convert pandas.dataframe to database table

In [ ]:
```
df_class.to_sql('class', conn, if_exists='replace', index=False)
df_student.to_sql('student', conn, if_exists='replace', index=False)
df_course.to_sql('course', conn, if_exists='replace', index=False)
df_score.to_sql('score', conn, if_exists='replace', index=False)
```

Out[ ]:
```
20
```

sqlite3 in python

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

Execute SQL statements

Select

```
In [ ]: sqlstr = 'select * from student'
        cur.execute(sqlstr)
```

```
Out[ ]: <sqlite3.Cursor at 0x1ab8bd84ac0>
```

Get all rows of the result

- cur.fetchone() returns the first row of the result

```
In [ ]: cur.fetchall()
```

```
Out[ ]: [(701001, '李小勇', '男', '1990-12-21', '南昌', '汉族', 'CS0701'),
         (701008, '王红', '男', '1992-04-26', '上海', '汉族', 'CS0701'),
         (703010, '李宏冰', '女', '1992-03-09', '太原', '蒙古族', 'AC0703'),
         (703045, '王红', '男', '1992-04-26', '北京', '汉族', 'AC0703'),
         (802002, '刘方晨', '女', '1990-11-11', '南昌', '傣族', 'ISO802'),
         (802005, '王红敏', '女', '1990-10-01', '上海', '蒙古族', 'ISO802')]
```

sqlite3 in python

Execute an insert statement

In [ ]:
```
sqlstr = '''insert into student values(450216,'李四','男', '1990-12-09', '北京', '傈僳族', 'EE0026')'''
cur.execute(sqlstr)
```

Out[ ]:
```
<sqlite3.Cursor at 0x1ab8bd84ac0>
```

Commit the changes

In [ ]:
```
conn.commit()
```

- Without this step, the changes cannot be updated.

Print the table

In [ ]:
```
cur.execute('select * from student').fetchall()
```

Out[ ]:
```
[(701001, '李小勇', '男', '1990-12-21', '南昌', '汉族', 'CS0701'),
 (701008, '王红', '男', '1992-04-26', '上海', '汉族', 'CS0701'),
 (703010, '李宏冰', '女', '1992-03-09', '太原', '蒙古族', 'AC0703'),
 (703045, '王红', '男', '1992-04-26', '北京', '汉族', 'AC0703'),
 (802002, '刘方晨', '女', '1990-11-11', '南昌', '傣族', 'ISO802'),
 (802005, '王红敏', '女', '1990-10-01', '上海', '蒙古族', 'ISO802'),
 (450216, '李四', '男', '1990-12-09', '北京', '傈僳族', 'EE0026')]
```

sqlite3 in python

Print the table

In [ ]:
```
cur.execute('select * from student').fetchall()
```

Out[ ]:
```
[(701001, '李小勇', '男', '1990-12-21', '南昌', '汉族', 'CS0701'),
 (701008, '王红', '男', '1992-04-26', '上海', '汉族', 'CS0701'),
 (703010, '李宏冰', '女', '1992-03-09', '太原', '蒙古族', 'AC0703'),
 (703045, '王红', '男', '1992-04-26', '北京', '汉族', 'AC0703'),
 (802002, '刘方晨', '女', '1990-11-11', '南昌', '傣族', 'ISO802'),
 (802005, '王红敏', '女', '1990-10-01', '上海', '蒙古族', 'ISO802'),
 (450216, '李四', '男', '1990-12-09', '北京', '傈僳族', 'EE0026')]
```

Convert database table into pandas.dataframe

In [ ]:
```
df_student = pd.read_sql('select * from student', conn)
df_student
```

Out[ ]:

	studentNo	studentName	sex	birthday	native	nation	classNo
0	701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
1	701008	王红	男	1992-04-26	上海	汉族	CS0701
2	703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
3	703045	王红	男	1992-04-26	北京	汉族	AC0703
4	802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
5	802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802
6	450216	李四	男	1990-12-09	北京	傈僳族	EE0026

sqlite3 in python

Close the connection

In [ ]:
```
cur.close()
conn.close()
```

Let’s try more SQL queries!

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentName FROM student;

studentName
李小勇
王红
李宏冰
王红
刘方晨
王红敏

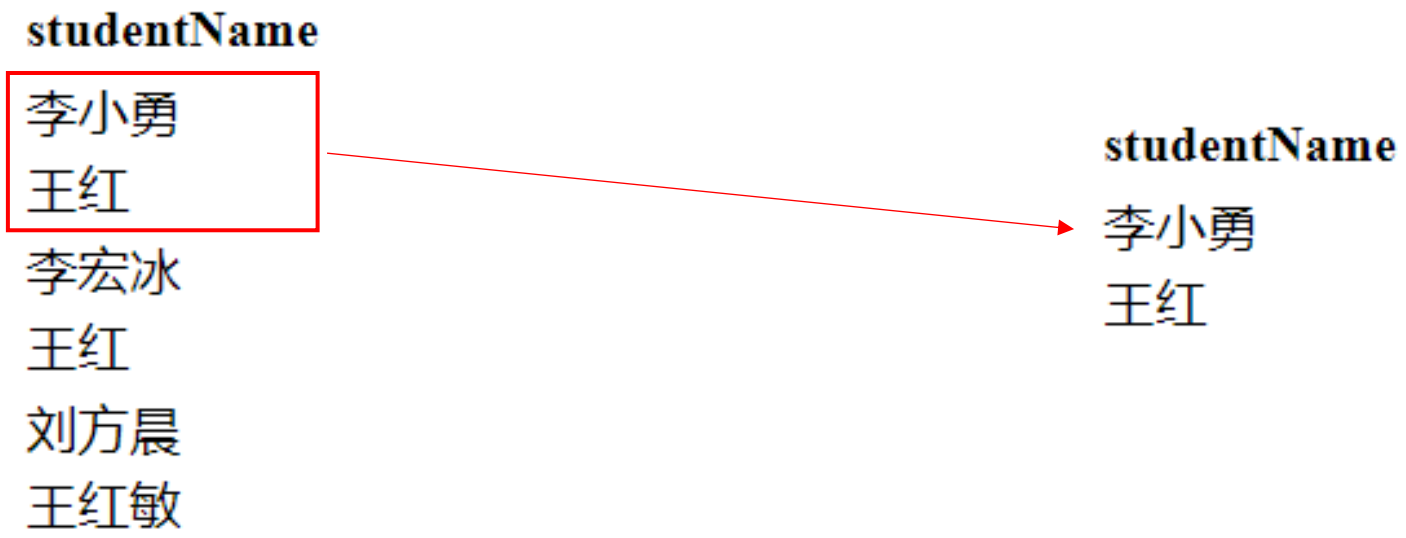
- SELECT DISTINCT studentName FROM student;

studentName
李小勇
王红
李宏冰
刘方晨
王红敏

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentName FROM student LIMIT 2;



Examples

- SELECT studentNo, score FROM score ORDER BY score LIMIT 5;

studentNo	score
703045	52
802002	72
703010	73
701008	76
703045	80

studentNo	courseNo	term	score
701001	CN028	7081	85
701001	CS012	7082	88
701001	CS015	8091	92
701008	AC001	7081	76
701008	CN028	7081	86
701008	CS012	7082	93
701008	CS015	8091	96
703010	AC001	7081	92
703010	CN028	7081	83
703010	CS012	7082	73
703045	AC001	7081	52
703045	AC001	8091	94
703045	CN028	7081	80
703045	CS015	8091	82
802002	AC001	8091	98
802002	CN028	8091	72
802002	CS015	9101	85
802005	AC001	9101	88
802005	CS012	8092	90
802005	CS015	9101	87

Examples

- SELECT studentNo, score FROM score ORDER BY score DESC LIMIT 5;

studentNo	score
802002	98
701008	96
703045	94
701008	93
701001	92

studentNo	courseNo	term	score
701001	CN028	7081	85
701001	CS012	7082	88
701001	CS015	8091	92
701008	AC001	7081	76
701008	CN028	7081	86
701008	CS012	7082	93
701008	CS015	8091	96
703010	AC001	7081	92
703010	CN028	7081	83
703010	CS012	7082	73
703045	AC001	7081	52
703045	AC001	8091	94
703045	CN028	7081	80
703045	CS015	8091	82
802002	AC001	8091	98
802002	CN028	8091	72
802002	CS015	9101	85
802005	AC001	9101	88
802005	CS012	8092	90
802005	CS015	9101	87

Examples

studentNo	courseNo	term	score
701001	CN028	7081	85
701001	CS012	7082	88
701001	CS015	8091	92
701008	AC001	7081	76
701008	CN028	7081	86
701008	CS012	7082	93
701008	CS015	8091	96
703010	AC001	7081	92
703010	CN028	7081	83
703010	CS012	7082	73
703045	AC001	7081	52
703045	AC001	8091	94
703045	CN028	7081	80
703045	CS015	8091	82
802002	AC001	8091	98
802002	CN028	8091	72
802002	CS015	9101	85
802005	AC001	9101	88
802005	CS012	8092	90
802005	CS015	9101	87

- SELECT studentNo, term, score FROM score ORDER BY term, score DESC LIMIT 10;

studentNo	term	score
703010	7081	92
701008	7081	86
701001	7081	85
703010	7081	83
703045	7081	80
701008	7081	76
703045	7081	52
701008	7082	93
701001	7082	88
703010	7082	73

Examples

studentNo	courseNo	term	score
701001	CN028	7081	85
701001	CS012	7082	88
701001	CS015	8091	92
701008	AC001	7081	76
701008	CN028	7081	86
701008	CS012	7082	93
701008	CS015	8091	96
703010	AC001	7081	92
703010	CN028	7081	83
703010	CS012	7082	73
703045	AC001	7081	52
703045	AC001	8091	94
703045	CN028	7081	80
703045	CS015	8091	82
802002	AC001	8091	98
802002	CN028	8091	72
802002	CS015	9101	85
802005	AC001	9101	88
802005	CS012	8092	90
802005	CS015	9101	87

- SELECT studentNo, score FROM score WHERE score BETWEEN 85 AND 90;

studentNo	score
701001	85
701001	88
701008	86
802002	85
802005	88
802005	90
802005	87

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, native FROM student WHERE studentName = '王红';

studentNo	studentName	native
701008	王红	上海
703045	王红	北京

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, native FROM student WHERE studentName != '王红';

studentNo	studentName	native
701001	李小勇	南昌
703010	李宏冰	太原
802002	刘方晨	南昌
802005	王红敏	上海

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, nation FROM student WHERE nation = '汉族' AND sex = '男';

studentNo	studentName	nation
701001	李小勇	汉族
701008	王红	汉族
703045	王红	汉族

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, nation FROM student WHERE nation = '傣族' OR nation = '蒙古族' ORDER BY nation;

studentNo	studentName	nation
802002	刘方晨	傣族
703010	李宏冰	蒙古族
802005	王红敏	蒙古族

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, nation FROM student WHERE nation IN ('傣族', '蒙古族') ORDER BY nation;

studentNo	studentName	nation
802002	刘方晨	傣族
703010	李宏冰	蒙古族
802005	王红敏	蒙古族

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, nation FROM student WHERE nation NOT IN ('傣族', '蒙古族') ORDER BY nation;

studentNo	studentName	nation
701001	李小勇	汉族
701008	王红	汉族
703045	王红	汉族

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, native FROM student WHERE studentNo LIKE '80%';

studentNo	studentName	native
802002	刘方晨	南昌
802005	王红敏	上海

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, native FROM student WHERE studentNo LIKE '70100_';

studentNo	studentName	native
701001	李小勇	南昌
701008	王红	上海

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, native FROM student WHERE studentNo REGEXP '01';

studentNo	studentName	native
701001	李小勇	南昌
701008	王红	上海
703010	李宏冰	太原

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, native FROM student WHERE studentNo REGEXP '.*01.*01.*';

studentNo	studentName	native
701001	李小勇	南昌

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

- SELECT studentNo, studentName, native FROM student WHERE studentNo REGEXP '.*0{2}.*';

studentNo	studentName	native
701001	李小勇	南昌
701008	王红	上海
802002	刘方晨	南昌
802005	王红敏	上海

Join – a powerful function for database

- SELECT $R_{i_1}, \ldots, R_{i_k}, S_{j_1}, \ldots, S_{j_l}$ FROM $r(R),\ s(S)$ WHERE $r.R_p = s.S_q$
- SELECT $R_{i_1}, \ldots, R_{i_k}, S_{j_1}, \ldots, S_{j_l}$ FROM $r(R)$ INNER JOIN $s(S)$ ON $r.R_p = s.S_q$

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

classNo	className	institute	grade	classNum
AC0703	会计学08(3)班	会计学院	2007	46
CS0701	计算机07(1)班	信息学院	2007	48
ISO802	信息系统08(2)班	信息学院	2008	43

- SELECT studentName, grade, nation FROM student, class WHERE student.classNo = class.classNo;
- SELECT studentName , grade, nation FROM student INNER JOIN class ON student.classNo = class.classNo;

studentName	grade	nation
李小勇	2007	汉族
王红	2007	汉族
李宏冰	2007	蒙古族
王红	2007	汉族
刘方晨	2008	傣族
王红敏	2008	蒙古族

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

classNo	className	institute	grade	classNum
AC0703	会计学08(3)班	会计学院	2007	46
CS0701	计算机07(1)班	信息学院	2007	48
ISO802	信息系统08(2)班	信息学院	2008	43

- SELECT studentName, grade, nation FROM student INNER JOIN class ON student.classNo = class.classNo WHERE class.grade = '2008' AND student.nation = '蒙古族';

studentName	grade	nation
李小勇	2007	汉族
王红	2007	汉族
李宏冰	2007	蒙古族
王红	2007	汉族
刘方晨	2008	傣族
王红敏	2008	蒙古族

View

- Some time-consuming queries may be repeated multiple times.
- CREATE VIEW V AS …

Examples

studentNo	studentName	sex	birthday	native	nation	classNo
701001	李小勇	男	1990-12-21	南昌	汉族	CS0701
701008	王红	男	1992-04-26	上海	汉族	CS0701
703010	李宏冰	女	1992-03-09	太原	蒙古族	AC0703
703045	王红	男	1992-04-26	北京	汉族	AC0703
802002	刘方晨	女	1990-11-11	南昌	傣族	ISO802
802005	王红敏	女	1990-10-01	上海	蒙古族	ISO802

classNo	className	institute	grade	classNum
AC0703	会计学08(3)班	会计学院	2007	46
CS0701	计算机07(1)班	信息学院	2007	48
ISO802	信息系统08(2)班	信息学院	2008	43

- CREATE VIEW student_class AS SELECT studentNo, studentName, sex, native,nation, class.* FROM student INNER JOIN class ON student.classNo = class.classNo;
- SELECT * FROM student_class;

studentNo	studentName	sex	native	nation	classNo	className	institute	grade	classNum
701001	李小勇	男	南昌	汉族	CS0701	计算机07(1)班	信息学院	2007	48
701008	王红	男	上海	汉族	CS0701	计算机07(1)班	信息学院	2007	48
703010	李宏冰	女	太原	蒙古族	AC0703	会计学08(3)班	会计学院	2007	46
703045	王红	男	北京	汉族	AC0703	会计学08(3)班	会计学院	2007	46
802002	刘方晨	女	南昌	傣族	ISO802	信息系统08(2)班	信息学院	2008	43
802005	王红敏	女	上海	蒙古族	ISO802	信息系统08(2)班	信息学院	2008	43

Further Reading

- CMU 15-445/645
  - https://15445.courses.cs.cmu.edu/fall2023/
  - Textbook: Database System Concepts (7th Edition)
- MySQL Crash Course (MySQL必知必会)
  - Forta, B. (2006). MySQL crash course. Pearson Education India.
- RisingWave
  - https://github.com/risingwavelabs/risingwave
  - https://mp.weixin.qq.com/s/WcOG2pJwjBBvvGDNWHF6eg

Q&A