Create a simple database with MySQL



1.Connect the MySQL server through MySQL Workbench

You can achieve many database operations by typing the SQL langue into the Query panel, such as creating a database, creating tables, selecting, updating, dropping tables or databases. This document is going to create a simple database instance for student management system in university.

2.Create a database

type the following SQL and click the execute (the third icon), Stu_Man is the name of created database. You will find the created database on the 'schemas' panel (left, if not, please refresh).



create database Stu_Man;

3.Create tables

MySQL is a typical relational database, which contains two basic concepts, entity and Relation (ER). Both entity and relation are described as tables, which are organized into rows and columns. All the data referenced by an attribute are in the same domain and conform to the same constraints. This instance contains three tables, that are tb_student (student information), tb_course (course information), and tb_student_course (the scores of student for course). We can describe them using the following ER-Diagram (<u>https://en.wikipedia.org/wiki/Entity-relationship_model</u>)

Rectangles: entities,

Diamonds: relationships

Circles: attributes

Underline means this attribute is the primary key which is the uniquely identifying attribute.



Create these tables by executing the following SQLs:

create table Stu_Man.tb_student

(

id int(11) not null auto_increment primary key, Name varchar(32) not null, Age int default 0,check(Age>0 and Age<=100), Gender boolean default 0,check(Gender=0 or Gender=1), Major varchar(32) not null);

create table Stu_Man.tb_course
(
 id int(5) not null auto_increment primary key,
Name varchar(32) not null,

```
Credit int(2)
);
create table Stu_Man.tb_student_course
(
id int(5) not null auto_increment primary key,
StudentID int(11),
CourseID int(5),
Score float(5)
);
```

tips: you can execute specific SQLs with selecting them, otherwise workbench will execute all the current SQLs in the query panel.



you will see your created tables (refresh):

▼ 🥃 Stu_Man	
Tables	
▶	
▶	
tb_student_course	
Views	

4.Import or insert (add) data into tables

Insert by executing SQL:

F Query 1 × F SQL File 3* × F tb_student ×
🖿 🖬 🖉 🖉 🙆 🕐 🧐 😨 Limit to 500 rows 🛛 🗿 🤧 🕑 🔍 🕦 📼
1 • Insert into Stu_Man.tb_student (id, Name, Age, Gender, Major)
2 values (001, "Marry", 21, 0, "Law");
3

Insert into Stu_Man.tb_student (id, Name, Age, Gender, Major) values (001, 'Marry', 21, 0, 'Law');

or using the visual grid:

select the table you want to edit, and click the grid icon (third) or right click the mouse and then "select...";



you will see the current data of this table, such as the record we just inserted by executing SQL:

100%	\$ 34	:1										
Result	Grid	. 4	Filter F	Rows:	Q Search	Edit: 👔	6 🖦 🖦	Export/Import:				
id	Name	Age	Gender	Major								
1	Marry	21	0	Law								Result
NULL	NULL	NULL	NULL	NULL								Gild
											_	
											_	•
											_	`
											_	
tb_s	tudent 1									Apply		Powert

you can achieve data edit (edit, insert, delete) or export and import data (many data formats are legal, such as .SQL, .CSV and so on).

5. One most important operation: "Select"

if I want to find out the students whose majors are "ComputerScience":

SELECT * *FROM Stu_Man.tb_student where Major="ComputerScience";* * means selecting all the attributes;

:	1 🛛 SELECT * FF	ROM Stu_	_Man.tl	b student where							
				1 🖸 SELECT * FROM Stu_Man.tb_student where Major="ComputerScience";							
100%	A 64·1										
100 /0											
		. Davies	0.0	and Table			_				
Resul	t Grid 📗 🛟 Filte	r Rows:	Q Se	earch Edit:	💪 🔜 🖶 Export/Import: 🏣 🐻						
id	It Grid 🔢 🚷 Filte	r Rows:	Q Se Gender	earch Edit: Major	🚄 誌 📪 Export/Import: 🏭 👸		Result				
id	It Grid 🔢 🔥 Filte Name John	r Rows: Age 22	Q Se Gender	earch Edit: Major ComputerScience	🖆 🔜 👪 🛛 Export/Import: 🏭 🐻		Result Grid				
id 2 3	t Grid III 🛟 Filte	r Rows: Age 22 22	Q Se Gender 1	earch Edit: Major ComputerScience ComputerScience	🖆 🔜 誌 🛛 Export/Import: 🎚 🐞		Result Grid				
id ≥ 2 3 9	tt Grid 🔢 🚷 Filte Name John Lee Liu	Age 22 22 21	Q Se Gender 1 1 1	Edit: Major ComputerScience ComputerScience ComputerScience	🖆 誌 📴 Export/Import: 🏭 🌇		Result Grid				
id 2 3 9	tt Grid III 🔅 Filte Name John Lee Liu	Age 22 22 21 NULL	Q Se Gender 1 1 1 NULL	Edit: Major ComputerScience ComputerScience ComputerScience	🖆 🔜 🏗 Export/Import: 🏭 🗞		Result Grid				
id 2 3 9 FULL	t Grid III 😯 Filte	Age 22 22 21	Q Se Gender 1 1 1 NULL	earch Edit: Major ComputerScience ComputerScience ComputerScience	🖆 🏗 📴 Export/Import: 🎚 👪		Result Grid				
id 2 3 9	tt Grid III 👬 🛟 Filte Name John Lee Liu	Age 22 22 21	Q Se Gender 1 1 1 NULL	Edit: Major ComputerScience ComputerScience ComputerScience	🖆 誌 📻 Export/Import: 🏭 🌇		Result Grid				
Hesur id 2 3 9 Rull	tt Grid III 👬 ᠢ Filte Name John Lee Liu	Age 22 22 21	Q Se Gender 1 1 1 NULL	Edit: Major ComputerScience ComputerScience ComputerScience	🖆 誌 🔚 Export/Import: 🏭 🇞		Result Grid				
Hesur id 2 3 9 Rott	It Grid III ᠢ Filte Name John Lee Liu	Age 22 22 21	Q Se Gender 1 1 1 1 NULL	Edit: Major ComputerScience ComputerScience ComputerScience	🖆 誌 🔚 Export/Import: 🏭 🇞		Result Grid				
Hesur id 2 3 9 RULL	tt Grid III 🔅 Filte Name John Lee Liu	Age 22 22 21	Q Se Gender 1 1 1 NULL	Edit: Major ComputerScience ComputerScience ComputerScience	🖆 📆 🔚 Export/Import: 🏭 🇞		Result Grid Form Editor				

If I just want to know their names:

SELECT Name FROM Stu_Man.tb_student where Major="ComputerScience";

Selecting from multiple tables with union: such as selecting students who are studying "Management"

course

🗀 🖬 💆 🖅 👰	🕑 🔂 🥏 🛞 🛛 Limit to 500 rows 🔹 🏡 🛫 🔍 🗻 🤕							
1 SELECT 2 Stu_Mar 3 where 1 4 and tb 5 and tb	<pre>SELECT tb_student.Name FROM Stu_Man.tb_student. Stu_Man.tb_student_course. Stu_Man.tb_course Where tb_student.id = tb_student_course.StudentID and tb_student_course.CourseID = tb_course.id and tb_course.Name = "Management":</pre>							
100% 🛟 35:5								
Result Grid 📗 🚷	Filter Rows: Q Search Export:							
Name								
Marry		Result Grid						
John								
Alice		Form						

SELECT tb_student.Name FROM Stu_Man.tb_student, Stu_Man.tb_student_course, Stu_Man.tb_course where tb_student.id = tb_student_course.StudentID and tb_student_course.CourseID = tb_course.id and tb_course.Name = "Management";

6. Advanced functions (statistics), group by and roll up

(http://dev.mysql.com/doc/refman/5.7/en/group-by-modifiers.html)

If I want to know the number of students for each course:



select Stu_Man.tb_student_course.CourseID, count(distinct(Stu_Man.tb_student_course.StudentID)) from Stu_Man.tb_student_course group by tb_student_course.CourseID;

If I want to know the number of students for each course and also the number of all the students:

🗲 tb_student × 🗲 SQL File 8* × 🗲 tb_student_course ×								
🗀 🖥 😿 💯 🐼 🕐 🔯 📀 🛞 🖉 🛛 Limit to 500 rows 💿 🗞 💅 🔍 🕦 🗨								
<pre>select Stu_Man.tb_student_course.CourseID, count(distinct(Stu_Man.tb_student_course.StudentID)) from Stu_Man.tb_student_course group by tb_student_course.CourseID with rollup;</pre>								
Result Grid 🔠 🚯 Filter Rows: 🔍 Q. Search 🔄 Export: 🔚								
CourseID count(distinct(Stu_Man.tb_student_course.S								
▶ 1 5								
2 4								
3 1								
NUL 5								

select Stu_Man.tb_student_course.CourseID, count(distinct(Stu_Man.tb_student_course.StudentID)) from Stu_Man.tb_student_course group by tb_student_course.CourseID with rollup; the difference is the last row in the second selection. The GROUP BY clause permits a WITH ROLLUP modifier that causes extra rows to be added to the summary output. These rows represent higher-level (or super-aggregate) summary operations. ROLLUP thus enables you to answer questions at multiple levels of analysis with a single query. It can be used, for example, to provide support for OLAP (Online Analytical Processing) operations.

Another example:

If I want to know the total scores, average scores for each student and the students from each major

<pre>I / fr Q O Q O O O I Limito 500 rows O</pre>								
100% 🛟 84:2	~	-		-	FIL.			-
Result Grid	💎 Filter	r Rows:	Q Search	Export:	82			
Major	studentID	sum_score	avg_score					
ComputerScience	2	19	9.5					
ComputerScience	3	18	9					
ComputerScience	NULL	37	9.25					
Law	1	24	8					
Law	NULL	24	8					_
Management	4	15	7.5					
Management	5	7	7					_
Management	NULL	22	7.333333333333333333333					
NULL	NULL	83	8.3					
Result 5								Bear

SELECT tb_student.Major, tb_student.id as studentID, sum(tb_student_course.Score) as sum_score,avg(tb_student_course.Score) as avg_score FROM Stu_Man.tb_student,Stu_Man.tb_student_course where tb_student.id = tb_student_course.StudentID group by tb_student.Major, tb_student.id with rollup;

More complicated SQLs (please refer http://www.w3schools.com/sql/,

http://dev.mysql.com/doc/refman/5.7/en/examples.html)

The referred database instance and another more complicated database for e-commerce

Download the example database in SQL format. You can import the complete database instance directly without building step by step.

Export database or a certain table:

Click the 'Data Export', and then select the target database or table, click the "Export to Self-Contained File" option, and finally click the "Start Export" button.

MANAGEMENT	∮ tb_student × ∮ SQL File 8* × ∮ tb_student_course ×	∲ tb_student × ∮ tb_student ×	Administration - Data Export ×	
Client Connections	Local instance 3306 Data Export			Advanced Options
I Status and System Variables Data Export Data Import/Restore	Tables to Export	Object Selection Export Progr	855	
INSTANCE I Starup / Shuddown Starup / Shuddown Starup / Shuddown Performance Performance Reports N Performance Schema Setup	Export Schema	Export Schema 2 b 2 b 2 b 2 b 2 b 2 b 2 b b 2 b b b 2 b b b b b b b b b b b b b b	Objects course student student_course	
CHEMAS	Refresh 3 tables selected	Dump Structure and Data	Select Views Select Tables	Unselect All
indiegogo indiegogo2 kiva	Objects to Export			
▶ prosper	Dump Stored Procedures and Functions	ump Events	Dump Triggers	
🔻 📄 Stu_Man	Export Options			
Tables Titles Views Stored Procedures Exerctions	Export to Dump Project Folder /Users/zhao Each table will be exported into a separate file. This allows a selective	hongke/dumps/Dump20161211 restore, but may be slower.		
▶ 📄 sys	 Export to Self-Contained File /Users/zhao 	hongke/Desktop/Untitled.sql		
	All selected database objects will be exported into a single, self-contri	ined file.		
	Create Dump in a Single Transaction (self-contained file	e only) Include Crea	te Schema	
Object Info Session Schema: Stu_Man	Press [Start Export] to start			Start Export

Import database or tables:

1. Download the referred database instance for student management

https://mega.nz/#!upZCUQBI!Wb9cwiv5yZ r0Nkns-SLBJWcdyAQc6UM6CvZPZ7YwQw

2. Create a new schema for the database, can be named as "Stu Man"





3. Set the created database as the default schema

▶ indiegogo▶ kiva	2	24 25 26	• p ¢	/*!401 CREATE `id`	01 SET TABLE	chara `tb_c	acter_s course` NULL A
▶ ⊜ prosper	Set as Default Sch	nema		`Nam	e` varo	har (32) NOT
▼ Stu_Ma ► Table Stu_Ma	Filter to This Sche Schema Inspector	ma	1 put	`Cre :1 ↓	dit` ir	nt(2)	DEFAUL
Store Func sys	Table Data Import Copy to Clipboard Send to SQL Edito	Wizard I ► or ►	ime	A	ction		
	Create Schema Alter Schema						
Object	Drop Schema						
Schema: Stu_I	Search Table Data						

4.Open the downloaded SQL file



5.Execute this SQL script and you will find the database on the left (refresh schemas).

Marging Market Ruging Scripting Help Control Control Control Control Control Control Control Contro	S MySQL Workbench	- 🗆 ×			
Test Edit Version Departs Regins Strights Height Control Note: Elevent Outyl Sub_Max Scheel Note: Scheel Note: Scheel Control Note: Elevent Outyl Sub_Max Scheel Note: Scheel Control Control Note: Elevent Outyl Sub_Max Scheel Note: Scheel Control Control Note: Elevent Test Note: Scheel Sub_Max Scheel Control Control Control Note: Scheel Test Note: Scheel Scheel Scheel Control Control Note: Scheel Test Note: Scheel Scheel Scheel Control Control Note: Scheel Test Note: Scheel Scheel Scheel Control Control Note: Scheel Test Note: Scheel Scheel Scheel Control Control Note: Scheel Test Note: Scheel Scheel Scheel Scheel Scheel Note: Scheel Test Note: Scheel Scheel Scheel Scheel Scheel Note: Scheel Test Note: Scheel Scheel Scheel Scheel Scheel Note: Scheel Test Note: Scheel Scheel Scheel Scheel Scheel No	SQL Editor (test) ×				
Control	File Edit View Query Database Plugins Scripting Help				
Openationality Openationality Openationality Openationality Openationality Control Control Control Control Control Control Control Control <td< th=""><th></th><th></th></td<>					
Control Control <t< th=""><th>Object Browser Ouerv 1 Stu Man - Schema Stu Man X</th><th>SOLAdditions</th></t<>	Object Browser Ouerv 1 Stu Man - Schema Stu Man X	SOLAdditions			
Puteriods Image: Strice Stri		🚸 🔅 My Stimuts 🔹 🖓			
Monutos Image:	Search objects	No is in a show 1140 c			
Winneds	▼ Stu_man 1 MySUL dump 10.13 Distrib 5.7.12, Tor 05x10.9 (X86_64)	^			
• D.g.maret • D.g.m	Tables 3 Host: localhost Database: Stu Man				
* Biginated conset 5	b th student 4				
* 10 Mona * * /*140011 STI @UD CWRACTER_ST_CLIENT=macromemotors ST CLIENT */; * init * * /*140011 STI @UD CWRACTER_ST_CLIENT=macromemotors ST RESULTS */; * /*140011 STI @UD CWRACTER_ST_CLIENT=macromemotors ST RESULTS */; * /*140011 STI @UD CWRACTER_ST_CLIENT=macromemotors ST RESULTS */; * /*140011 STI @UD CWRACTER_ST_CLIENT=macromemotors ST RESULTS */; * /*140011 STI @UD CWRACTER_ST_CLIENT=SWC:NUTCONCOMECTION */; * * * * * * * * * * * * * * * * * * *	b_student_course 5 Server version 5.7.13				
* Test *** /*********************************	Winds A A				
• • / **10010 SET #000_OULATION_CONNECTION=@@OULATION_CONNECTION */; * * /*10010 SET MANE_UTE */; * /*10011 SET MONE_MONE_MONE_MONE_WONE_MONE_WONE_WONE_WONE_WONE_WONE_WONE_WONE_W	▶ ettest s • /*!40101 SET @OLD CHARACTER SET RESULTS=@@CHARACTER SET RESULTS=	7:			
	9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;				
if a // 140103 \$11 (MOULD THE_ZONE - 26) (MIC) if a // 140103 \$11 (MOULD THE ZONE - 26) (MIC) if a // 140103 \$11 (MOULD THE ZONE - 26) (MIC) if a // 140103 \$11 (MOULD THE ZONE - 26) (MIC) if a // 140101 \$11 (MIC) THE ZONE - 260 (MIC) if a // 140101 \$11 (MIC) THE ZONE - 260 (MIC) if a // 140101 \$11 (MIC) THE ZONE - 260 (MIC) if a // 140101 \$11 (MIC) if a // 140101 if a // 140101 if a // 140101 if a // 140101 if a // 14010 if a /	10 • /*!40101 SET NAMES utf8 */;				
	11 • /*!40103 SET @OLD TIME_ZONE=@01ME_ZONE */;				
Sofematics is * /*ieonit SET goll_OF CHECKS = gold_CHECKS = Gold_KEY /*ieonit SET gold_SOL_MOTES, SOL_MOTES, SO	12 • /*:40105 SET TIME TONE +00:00 -7;	*/.			
is * /**140101 SET @010_SUL_MODE=w50L_MODE=*0.AUTO_VALUE_ON_ZERO is * /**140101 SET @010_SUL_MODE=w50L_MODE>*0.Y; if * /**140101 SET @010_SUL_MODE>*0.*/S if * /**********************	14 • /*!40014 SET @OLD FOREIGN KEY CHECKS=@@FOREIGN KEY CHECKS, FOREIG	SN KEY			
Sofematics 35 * (*14011) SET @0LD_SQL_NOTES*@000L_NOTES, SQL_NOTES*@ */; 36	15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON	V_ZER0			
Advances	16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;				
Operation	17				
Advances	19 Table structure for table `tb course`				
21 00ptc1/0 0000 TABLE IF EXISTS 'tb course'; 23 * (**********************************	20				
22 • (Deb) TABLE IF EXISTS 'to course'; 24 • (Field) IE (Baved (course); 24 • (Field) IE (Baved (course); 24 • (Course); 24 • (Field) IE (Baved (course); 24 • (Course); 25 • (Field) IE (Baved (course); 25 • (Course); 27	21				
23 # // Hollin Stig Balved & Client * WCharder St Client */; 24 minutedia 25 minutedia 26 minutedia 26 minutedia 26 minutedia 26 minutedia 27 minutedia 28 minutedia 29 minutedia 29 minutedia 29 minutedia 29 minutedia 20 minutedia <th>22 • DROP TABLE IF EXISTS 'tb course';</th> <th></th>	22 • DROP TABLE IF EXISTS 'tb course';				
Advances Sector 22 Concernsor 22 <th>23 • /*140101 SET (saved cs client = @dcharacter.set client */;</th> <th></th>	23 • /*140101 SET (saved cs client = @dcharacter.set client */;				
Intermedia 26 1/10* // 10* <th 10*<="" th=""> // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* <th 10*<="" th=""> <th 10*<="" th=""> // 10*</th><th>25 • FICRATE TABLE 'th course' (</th><th></th></th></th>	// 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* // 10* <th 10*<="" th=""> <th 10*<="" th=""> // 10*</th><th>25 • FICRATE TABLE 'th course' (</th><th></th></th>	<th 10*<="" th=""> // 10*</th> <th>25 • FICRATE TABLE 'th course' (</th> <th></th>	// 10*	25 • FICRATE TABLE 'th course' (
Schema: Nu, man 27 Name" varchar(32) NOT NULL, Snoph Operative Image: Name" varchar(32) NOT NULL, Snoph Action Coput Image: Name" varchar(32) NOT NULL, Snoph Image: Name varchar(32) NOT NULL, Snoph Snoph Image: Name varchar(32) NOT NULL, Image: Name varchar(32) NOT NULL, Snoph Image: Name varchar(32) NOT NULL, Oment/Image: Name varchar(32) NOT NULL, Development/Image: Name varchar(32) Not Null, Image: Name varchar(32) Not Null, Image: Name varchar(32) Not Null, Oment/Image: Name varchar(32) Not Null, Development/Image: Name varchar(32) Null, Image: Name varchar(32) Null, Image: Name varchar(32) Null, Oment/Image: Name varchar(32) Null, Development/Image: Name varchar(32) Null, Development/Image: Name varchar(32) Null, Development/Image: Name varchar(32) Null, Development varc	Information 26 id' int(5) NOT NULL AUTO_INCREMENT,				
Optiont	Schema: stu_man 27 `Name` varchar(32) NOT NULL,	 Snippets 			
Actor Order Image Description 65:112.64 Actor Order 0 Description	Output				
Three Acces Manage Downson / Teths A 0 69 112540 /M014 SET UNDUE_CHEDIG=@OLD_UNDUE_CHEDIG="0">Umage Downson / Teths A 0 70 112540 /M0101 SET CHARACTER_SET_CL_DHAT=@OLD_CHARACTER_SET_CL_ Downson / Method 0.000 eec 0 71 112540 /M0101 SET CHARACTER_SET_CL_OHARACTER_SET_CL_ Devolgi Method 0.000 eec 0 71 112540 /M0101 SET CHARACTER_SET_DESUT=% Downson / Method 0.000 eec 0 71 112540 /M0101 SET CHARACTER_SET_DESUT=% Downson / Method 0.000 eec 0 71 112540 //M0111 SET SOL_NOTES- Downson / Method 0.000 eec 0 71 112540 //M0111 SET SOL_NOTES-@ 0 0 0 0000 eec 0	Action Output -				
0 61 112540 AVX011 SET UNX012_CPECKCS * 00 UNX012_EPECKS Y 0 unx01 Method 0.000 ecc 0 71 112540 AVX011 SET UNX012_EPECKS * 00 UNX012_EPECKS Y 0 unx01 Method 0.000 ecc 0 71 112540 AVX011 SET COULARCELES, SET LESUITS-RED.D_LANARCELES, SET LESUITS-RED.D_LOXAL_XOTIES 0 unx01 Method 0.000 ecc 0 72 112540 AVX011 SET COULATION_LOXARECTOR-REGULE_COULATION_LOXANEE 0 unx01 Method 0.000 ecc 0 73 112540 AVX011 SET SOUL_SOUL_SOUL_SOUL_SOUL_SOUL_SOUL_SOUL_	Time Action Message	Duration / Fetch			
0 70 112540 //40101 SET CHARACTER, SET CLUENT-@OLD_CHARACTER, SET_C. 0 monyla Meteod 0.000 eec 0 71 112540 /40101 SET CHARACTER, SET CLUENT-@OLD_CHARACTER, SET_C. 0 monyla Meteod 0.000 eec 0 72 112540 /40101 SET CHARACTER, SET_C. 0 monyla Meteod 0.000 eec 0 73 112540 /40101 SET SQL_NOTES-@OLD_SQL_NOTES / 0 monyla Meteod 0.000 eec 0 73 112540 /40111 SET SQL_NOTES-@OLD_SQL_NOTES / 0 monyla Meteod 0.000 eec 0 73 112540 /40111 SET SQL_NOTES-@OLD_SQL_NOTES / 0 monyla Meteod 0.000 eec 0 73 112540 /40111 SET SQL_NOTES-@OLD_SQL_NOTES / 0 monyla Meteod 0.000 eec 0 73 112540 /40111 SET SQL_NOTES-@OLD_SQL_NOTES / 0 monyla Meteod 0 000 eec v	69 11.25:40 /*140014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS / 0 row(s) affected	0.000 sec			
0 71 112540 //MIDI SET CHARACTER, SET, JESUIT-SeqUID, CAMANCTER, SET, Di wondy Hindred 0.000 ecc 0 72 112540 //MIDI SET COLLANDI, CONDITION-ROLD, COLLATION, CONIEL. 0 wondy Hindred 0.000 ecc 0 72 112540 //MIDI SET SOL, NOTES-QUID, COLLATION, CONIEL. 0 wondy Hindred 0.000 ecc 0 73 112540 //MIDI SET SOL, NOTES-QUID, SOL, NOTES- 0 wondy Hindred 0.000 ecc Colynet Mid 5 session 73 112540 //MIDI SET SOL, NOTES-QUID, SOL, NOTES- 0 wondy Hindred 0.000 ecc Colynet Mid 5 session 73 112540 //MIDI SET SOL, NOTES-QUID, SOL, NOTES- 0 wondy Hindred 0.000 ecc	O 11:25:40 /*140101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_C 0 row(s) affected	0.000 sec			
© 72 112540 /A0101 SET COLLATON_CONNECTION=@OLL_COLLATON_CONNECT ON=@OLL_COLLATON_CONNECT ON=@OLLATON_CONN	71 11:25:40 /*140101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET 0 row(s) affected	0.000 sec			
Chine Season Chine Comparison Company Company and Company	72 11:25:40 /*140101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNE 0 row(s) affected	0.000 sec			
Object Info Session V Query Completed I	73 11:25:40 /140111 SET SQL_NOTES * @OLD_SQL_NOTES */ 0 row(s) affected	0.000 sec			
Query Completed [2]	Object Info Session	·			
	Query Completed				

A more complicated database instance for e-commerce:

1.Download the complicated instance

https://mega.nz/#!XtJigL6T!TOc-xg4e3bVXYXwoOiuMfKjxYBgmkLTjOqnmyrfnPjg and import this database in the similar way (above).

2.ER-diagram for this data



3.Data description

this database contains four tables, i.e., user, item, merchant_inf and behavior

Table name	user	item	Merchant_inf	behavior
#column	2	5	2	5
#row	49	100	247	980

Some attribute descriptions:

User table: the user information

User_id,

Name, (the user name, which is private and invisible)

Item table: the item information

Id,

Name,

Price,

Category, Category_id,

Merchant_inf table: the merchant information Merchant_id, Budget, budget constraints imposed on the merchant

Behavior table: users' behaviors, such as "click or buy" some items User_id, Merchant_id, Time_stamp, the time of behavior Item_id, Online_Action_id, "0" denotes "click" while "1" for "buy"