Exercises: Functions, Triggers and Transactions

This document defines the **exercise assignments** for the "Databases Basics - MySQL" course @ Software University.

Part I – Queries for SoftUni Database

Problem 1. Employees with Salary Above 35000

Create stored procedure **usp_get_employees_salary_above_35000** that returns **all employees' first and last names** for whose **salary is above 35000**. Submit your query statement as Run skeleton, run queries & check DB in Judge.

Example

first_name	last_name
Roberto	Tamburello
David	Bradley
Terri	Duffy
…	…

Problem 2. Employees with Salary Above Number

Create stored procedure **usp_get_employees_salary_above** that **accept a number** as parameter and return **all employees' first and last names** whose salary is **above or equal** to the given number. Submit your query statement as Run skeleton, run queries & check DB in Judge.

Example

Supplied number for that example is 48100.

first_name	last_name
Terri	Duffy
Jean	Trenary
Ken	Sanchez
…	…

Problem 3. Town Names Starting With

Write a stored procedure **usp_get_towns_starting_with** that **accept string as parameter** and returns **all town names starting with that string.** Submit your query statement as Run skeleton, run queries & check DB in Judge.

Example

Here is the list of all towns **starting with "b".**

town
Bellevue
Bothell
Bordeaux
Berlin

Problem 4. Employees from Town

Write a stored procedure **usp_get_employees_from_town** that accepts **town name** as parameter and return the **employees' first and last name that live in the given town.** Submit your query statement as Run skeleton, run queries & check DB in Judge.

Example

Here it is a list of employees **living in Sofia.**

first_name	last_name
Svetlin	Nakov
Martin	Kulov
George	Denchev

Problem 5. Salary Level Function

Write a function **ufn_get_salary_level** that receives **salary of an employee** and returns the **level of the salary**.

- If salary is **< 30000** return **"Low"**
- If salary is **between 30000 and 50000 (inclusive)** return **"Average"**
- If salary is **> 50000** return **"High"**

Submit your query statement as Run skeleton, run queries & check DB in Judge.

Example

salary	salary_Level
13500.00	Low
43300.00	Average
125500.00	High

Problem 6. Employees by Salary Level

Write a stored procedure **usp_get_employees_by_salary_level** that receive as **parameter level of salary** (low, average or high) and print the **names of all employees** that have given level of salary.

Example

Here is the list of all employees with **high salary**.

first_name	last_name
Terri	Duffy
Jean	Trenary
Ken	Sanchez
…	…

Problem 7. Define Function

Define a function **ufn_is_word_comprised(set_of_letters, word)** that returns **true** or **false** depending on that if the word is a comprised of the given set of letters. Submit your query statement as Run skeleton, run queries & check DB in Judge.

Example

set_of_letters	word	result
oistmiahf	Sofia	1
oistmiahf	halves	0
bobr	Rob	1
pppp	Guy	0

Problem 8. * Delete Employees and Departments

Write a SQL query to delete all Employees from the **Production** and **Production Control** departments. Also **delete these departments from the Departments table.** After that exercise restore your database to revert those changes.

PART II – Queries for Bank Database

Problem 9. Find Full Name

You are given a database schema with tables **account_holders(id (PK), first_name, last_name, ssn)** and **Accounts(id (PK), account_holder_id (FK), balance)**. Write a stored procedure **usp_get_holders_full_name** that selects the full names of all people. Submit your query statement as Run skeleton, run queries & check DB in Judge.

Example

full_name
Susan Cane
Kim Novac
Jimmy Henderson
…

Problem 10. People with Balance Higher Than

Your task is to create a stored procedure **usp_get_holders_with_balance_higher_than** that accepts a **number as a parameter** and returns all **people who have more money in total of all their accounts than the supplied number**. Submit your query statement as Run skeleton, run queries & check DB in Judge.

Example

first_name	last_name
Susan	Cane
Petar	Kirilov
…	…

Problem 11. Future Value Function

Your task is to create a function **ufn_calculate_future_value** that accepts as parameters – **sum**, **yearly interest rate** and **number of years**. It should calculate and return the future value of the initial sum. Using the following formula:

$$FV = I \times ((1 + R)^T)$$

- **I** – Initial sum
- **R** – Yearly interest rate
- **T** – Number of years

Submit your query statement as Run skeleton, run queries & check DB in Judge.

Example

Input	Output
Initial sum: 1000 **Yearly Interest rate:** 10% **years:** 5 `ufn_calculate_future_value(1000, 0.1, 5)`	`1610.51`

Problem 12. Calculating Interest

Your task is to create a stored procedure **usp_calculate_future_value_for_account** that uses the function from the previous problem to give an interest to a person's account **for 5 years**, along with information about his/her **account id, first name, last name and current balance** as it is shown in the example below. It should take the **account_Id** and the **interest rate** as parameters.

Example

account_id	fist_name	last_name	current_balance	balance_in_5_years
1	Susan	Cane	123.12	198.286

Problem 13. Deposit Money

Add stored procedure **usp_deposit_money** (account_id, money_amount) that operate in transactions. Submit your query statement as Run skeleton, run queries & check DB in Judge.

Problem 14. Withdraw Money

Add stored procedures **usp_withdraw_money** (account_id, money_amount) that operate in transactions. Submit your query statement as Run skeleton, run queries & check DB in Judge.

Problem 15. Money Transfer

Write stored procedure that **transfers money from one account to another**. Consider cases when one of the **account_ids** is not valid or the amount of **money is negative number**. Make sure that the whole procedure **passes without errors** and **if error occurs make no change in the database.**

Problem 16. Create Table Logs

Create another table – **logs** (log_id, account_id, old_sum, new_sum). Add a trigger to the accounts table that enters a new entry into the Logs table every time the sum on an account changes.

Example

log_id	account_id	old_sum	new_sum
1	1	123.12	113.12
…	…	…	…

Problem 17. Create Table Emails

Create another table – **notification_emails**(id, recipient, subject, body). Add a trigger to logs table and **create new email whenever new record is inserted in logs table.** The following data is required to be filled for each email:

- **recipient** – account_id
- **subject** – "Balance change for account: **{account_id}**"
- **body** - "On **{date}** your balance was changed from **{old}** to **{new}.**"

Example

id	recipient	subject	body
1	1	Balance change for account: 1	On 2016-09-15 11:44:06 your balance was changed from 133 to 143.
…	…	…	…

PART III – Queries for Diablo Database

You are given a **database "Diablo"** holding users, games, items, characters and statistics available as SQL script. Your task is to write some stored procedures, views and other server-side database objects and write some SQL queries for displaying data from the database.

Important: start with a **clean copy of the "Diablo" database on each problem**. Just execute the SQL script again.

Problem 18. Trigger

Users should not be allowed to buy items with higher level than their level. Create a trigger that restricts that.

Add bonus cash of 50000 to users: **baleremuda, loosenoise, inguinalself, buildingdeltoid, monoxidecos** in the game **"Bali".**

There are two groups of items that you should buy for the above users in the game. First group is with **id between 251 and 299 including**. Second group is with **id between 501 and 539 including.**

Take off cash from each user for the bought items.

Select all users in the current game with their items. Display **username**, **game name**, **cash** and **item name**. Sort the result by username alphabetically, then by item name alphabetically.

Output

Username	Name	Cash	Item Name
baleremuda	Bali	4****.**	Iron Wolves Doctrine
baleremuda	Bali	4****.**	Irontoe Mudsputters
…	…	…	…
buildingdeltoid	Bali	3****.**	Alabaster Gloves
…	…	…	…

Problem 19. Massive Shopping

1. User **Stamat** in **Safflower** game wants to buy some items. He likes all items **from Level 11 to 12** as well as all items **from Level 19 to 21.** As it is a bulk operation you have to **use transactions.**
2. A transaction is the operation of taking out the cash from the user in the current game as well as adding up the items.
3. Write transactions for each level range. If anything goes wrong turn back the changes inside of the transaction.
4. Extract all item names in the given game sorted by name alphabetically

Output

Item Name
Akarats Awakening
Amulets
Angelic Shard
…

Problem 20. Number of Users for Email Provider

Find number of users for email provider from the largest to smallest, then by Email Provider in ascending order. Submit your query statement as Prepare DB & run queries in Judge.

Output

Email Provider	Number of Users
yahoo.com	14
dps.centrin.net.id	5
softuni.bg	5
indosat.net.id	4
…	…

Problem 21. All User in Games

Find all **user in games** with information about them. Display the game name, game type, username, level, cash and character name. Sort the result by level in descending order, then by username and game in alphabetical order. Submit your query statement as Prepare DB & run queries in Judge.

Output

Game	Game Type	Username	Level	Cash	Character
Calla lily white	Kinky	obliquepoof	99	7527.00	Monk
Dubai	Funny	rateweed	99	7499.00	Barbarian
Stonehenge	Kinky	terrifymarzipan	99	4825.00	Witch Doctor
…		…	…	…	…

Problem 22. Users in Games with Their Items

Find all users in games with their items count and items price. Display the username, game name, items count and items price. Display only user in games with items count more or equal to 10. Sort the results by items count in descending order then by price in descending order and by username in ascending order. Submit your query statement as Prepare DB & run queries in Judge.

Output

Username	Game	Items Count	Items Price
skippingside	Rose Fire & Ice	23	11065.00
countrydecay	Star of Bethlehem	18	8039.00
obliquepoof	Washington D.C.	17	5186.00

Problem 23. * User in Games with Their Statistics

Find all users in games with their statistics. Display the username, game name, character name, strength, defense, speed, mind and luck. Every statistic (strength, defense, speed, mind and luck) should be a sum of the character statistic, game type statistic and items for user in game statistic. Order the results by Strength in descending order, then by Defense in descending order, then by Speed in descending order, then by Mind in descending order, then by Luck in descending order. Submit your query statement as Prepare DB & run queries in Judge.

Output

Username	Game	Character	Strength	Defense	Speed	Mind	Luck
skippingside	Rose Fire & Ice	Sorceress	258	215	246	240	263
countrydecay	Star of Bethlehem	Sorceress	221	163	216	153	196
obliquepoof	Washington D.C.	Paladin	204	200	183	185	185

Problem 24. All Items with Greater than Average Statistics

Find all items with statistics larger than average. Display only items that have **Mind, Luck** and **Speed** greater than average **Items** mind, luck and speed. Sort the results by item names in alphabetical order.

Output

Name	Price	MinLevel	Strength	Defence	Speed	Luck	Mind
Aether Walker	473.00	46	2	10	15	11	13
Ancient Parthan Defenders	566.00	38	5	7	10	19	18
Aquila Cuirass	405.00	76	5	7	10	19	18
Arcstone	613.00	50	2	10	15	11	13

Problem 25. Display All Items with information about Forbidden Game Type

Find **all items** and information whether and what forbidden game types they have. Display item name, price, min level and forbidden game type. Display all items. Sort the results by game type in descending order, then by item name in ascending order. Submit your query statement as Prepare DB & run queries in Judge.

Output

Item	Price	MinLevel	Forbidden Game Type
Archfiend Arrows	531.00	8	Kinky
Behistun Rune	611.00	67	Kinky
Boots	782.00	44	Kinky
…	…	…	…

Problem 26. Buy Items for User in Game

1. User **Alex** is in the shop in the game "**Edinburgh**" and she wants to buy some items. She likes **Blackguard**, **Bottomless Potion of Amplification**, **Eye of Etlich (Diablo III)**, **Gem of Efficacious Toxin**, **Golden Gorget of Leoric** and **Hellfire Amulet**. Buy the items. You should add the data in the right tables. Get the money for the items from user in game **Cash**.
2. Select all users in the current game with their items. Display username, game name, cash and item name. Sort the result by item name.

Create stored procedure **usp_buy_items_for_alex ()** that combines subtasks 1 and 2. Submit your procedure as Run skeleton, Run queries & check DB in Judge.

Output

Username	Name	Cash	Item Name
Alex	Edinburgh	****.**	Akanesh, the Herald of Righteousness
…	…	…	…
corruptpizz	Edinburgh	****.**	Broken Crown
…	…	…	…
printerstencils	Edinburgh	****.**	Envious Blade

PART IV – Queries for Geography Database

Problem 27. Peaks and Mountains

Find all **peaks along with their mountain** sorted by elevation (from the highest to the lowest), then by peak name alphabetically. Display the peak name, mountain range name and elevation. Submit your query statement as Prepare DB & run queries in Judge.

Output

PeakName	Mountain	Elevation
Everest	Himalayas	8848
K2	Karakoram	8611
Kangchenjunga	Himalayas	8586
…		…

Problem 28. Peaks with Their Mountain, Country and Continent

Find all peaks along with their mountain, country and continent. When a mountain belongs to multiple countries, display them all. Sort the results by peak name alphabetically, then by country name alphabetically. Submit your query statement as Prepare DB & run queries in Judge.

Output

PeakName	Mountain	CountryName	ContinentName
Aconcagua	Andes	Argentina	South America
Aconcagua	Andes	Chile	South America
Banski Suhodol	Pirin	Bulgaria	Europe
…	…	…	…

Problem 29. Rivers by Country

For each country in the database, display the number of rivers passing through that country and the total length of these rivers. When a country does not have any river, display **0** as rivers count and as total length. Sort the results by rivers count (from largest to smallest), then by total length (from largest to smallest), then by country alphabetically. Submit your query statement as Prepare DB & run queries in Judge.

Output

CountryName	ContinentName	RiversCount	TotalLength
China	Asia	8	35156
Russia	Europe	6	26427
…		…	…

Problem 30. Count of Countries by Currency

Find the **number of countries for each currency**. Display three columns: currency code, currency description and number of countries. Sort the results by number of countries (from highest to lowest), then by currency description alphabetically. Name the columns exactly like in the table below. Submit your query statement as Prepare DB & run queries in Judge.

Output

CurrencyCode	Currency	NumberOfCountries
EUR	Euro Member Countries	35
USD	United States Dollar	17
AUD	Australia Dollar	8
XOF	Communauté Financière Africaine (BCEAO) Franc	8
…	…	…

Problem 31. Population and Area by Continent

For each continent, display the total area and total population of all its countries. Sort the results by population from highest to lowest. Submit your query statement as Prepare DB & run queries in Judge.

Output

ContinentName	CountriesArea	CountriesPopulation
Asia	31603228	4130318467
Africa	30360296	1015470588
…	…	…

Problem 32. Monasteries by Country

1. Create a **table monasteries(id, name, country_code)**. Use auto-increment for the primary key. Create a **foreign key** between the tables Monasteries and Countries.
2. Execute the following SQL script (it should pass without any errors):

```
INSERT INTO monasteries(name, country_code) VALUES
('Rila Monastery “St. Ivan of Rila”', 'BG'),
('Bachkovo Monastery “Virgin Mary”', 'BG'),
('Troyan Monastery “Holy Mother''s Assumption”', 'BG'),
('Kopan Monastery', 'NP'),
('Thrangu Tashi Yangtse Monastery', 'NP'),
('Shechen Tennyi Dargyeling Monastery', 'NP'),
('Benchen Monastery', 'NP'),
('Southern Shaolin Monastery', 'CN'),
('Dabei Monastery', 'CN'),
('Wa Sau Toi', 'CN'),
('Lhunshigyia Monastery', 'CN'),
('Rakya Monastery', 'CN'),
('Monasteries of Meteora', 'GR'),
('The Holy Monastery of Stavronikita', 'GR'),
('Taung Kalat Monastery', 'MM'),
('Pa-Auk Forest Monastery', 'MM'),
('Taktsang Palphug Monastery', 'BT'),
('Sümela Monastery', 'TR')
```

3. Write a SQL command to add a new Boolean column **`is_deleted`** in the **`countries`** table (default is **`false`**). Note that there is no "Boolean" type in MySQL, so you should use an alternative.
4. Write and execute a SQL command to **mark as deleted all countries that have more than 3 rivers**.
5. Write a query to display all **monasteries** along with their **countries** sorted by monastery name. Skip all deleted countries and their monasteries.

Submit your query statements **only for subtasks 1, 2, 4 and 5 at once** as Prepare DB & run queries in Judge.

Output

Monastery	Country
Bachkovo Monastery “Virgin Mary”	Bulgaria
Benchen Monastery	Nepal
Kopan Monastery	Nepal
…	…

Problem 33. Monasteries by Continents and Countries

This problem assumes that the previous problem is completed successfully without errors.

1. Write and execute a SQL command that **changes the country named "Myanmar" to its other name "Burma"**.
2. Add a **new monastery** holding the following information: Name="**`Hanga Abbey`**", Country="**`Tanzania`**".
3. Add a **new monastery** holding the following information: Name="**`Myin-Tin-Daik`**", Country="**`Myanmar`**".
4. Find the **count of monasteries for each continent and not deleted country**. Display the **continent name**, the **country name** and the **count of monasteries**. Include also the countries with 0 monasteries. Sort the results by monasteries count (from largest to lowest), then by country name alphabetically. Name the columns exactly like in the table below.

Submit all your query statements at once as Prepare DB & run queries in Judge.

Output

ContinentName	CountryName	MonasteriesCount
Asia	Nepal	4
Europe	Bulgaria	3
Asia	Burma	2
Europe	Greece	2
…	…	…