

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту



Лабораторна робота 1
з організації баз даних та знань

Виконав:

Студент групи КН-208

Воробель Адріан

Викладач:

Якимишин Х.М.

Львів – 2019р.

Мета роботи: Розробити SQL запити для виконання операцій реляційної алгебри: об'єднання, перетину, різниці, декартового добутку.

Короткі теоретичні відомості.

Реляційна алгебра – це множина операцій, що виконуються над відношеннями і мають за мету утворення нових відношень або їх станів. Реляційна алгебра визначає операції, які однаковим чином реалізуються в усіх базах даних реляційного типу, незалежно від їх змісту і технологій, за допомогою яких вони реалізовані. Тобто реляційна алгебра представляє собою процедурну мову обробки реляційних таблиць.

Реляційна алгебра складається з таких операцій: об'єднання, перетин, різниця, декартовий добуток, проекція, селекція, натуральне з'єднання, умовне з'єднання, а також операції включення/вилучення кортежу з відношень, включення/вилучення атрибуту з відношення, зміни параметрів атрибуту. Перші чотири операції взяті з математичної теорії множин і практично співпадають з операціями над множинами. Це зручно, оскільки реляційні таблиці є множинами, і цілком природно застосовувати до них операції над множинами.

Об'єднанням двох відношень R та S з відповідними множинами атрибутів (A_1, A_2, \dots, A_n) називається відношення T , що має ту саму множину атрибутів (A_1, A_2, \dots, A_n) , а

його інформаційне наповнення утворюється кортежами першого та другого відношень за вилученням повторень:

$$R \cup S = T(A_1, A_2, \dots, A_n) = \{r\} \cup \{s\}$$

Об'єднання дозволяє нам комбінувати дані з двох таблиць з однаковими множинами атрибутів. Однакові множини атрибутів потрібні для того, щоб результатом виконання операції об'єднання була реляційна таблиця.

Перетином двох відношень R та S з відповідними множинами атрибутів (A_1, A_2, \dots, A_n) називається відношення T , що має ту саму множину атрибутів (A_1, A_2, \dots, A_n) , а

його інформаційне наповнення утворюється кортежами, які є спільними для цих двох відношень:

$$R \cap S = T(A_1, A_2, \dots, A_n) = \{r\} \cap \{s\}$$

Операція перетину дозволяє нам ідентифікувати рядки, спільні для двох таблиць.

Різницею двох відношень R та S з відповідними множинами атрибутів (A_1, A_2, \dots, A_n) називається відношення T , що має ту саму множину атрибутів (A_1, A_2, \dots, A_n) , а його інформаційне наповнення утворюється кортежами першого відношення за вилученням кортежів, які є спільними з другим відношенням:

$$R \setminus S = T(A_1, A_2, \dots, A_n) = \{r\} \setminus \{s\}$$

Операція різниці дозволяє ідентифікувати ті рядки, які є в одній таблиці, але відсутні в іншій.

Декартовим добутком двох відношень R та S з відповідними множинами атрибутів

(A_1 , A_2 ,..., A_n) та (B_1 , B_2 ,..., B_m) називається нове відношення T , множина атрибутів якого складається з об'єднання множини атрибутів двох відношень, а кожен кортеж інформаційного наповнення утворюється шляхом конкатенації (сполучення) кожного кортежу першого відношення з кожним кортежем другого відношення.

Для реалізації теоретико-множинних операцій на мові SQL використовують директиву SELECT, спрощений опис якої наведено далі, а також функції роботи з множинами значень IN(), NOT IN().

SELECT

[ALL | DISTINCT | DISTINCTROW]

елемент_вибірки [, *елемент_вибірки*] [**FROM** *перелік_таблиць*]

[**WHERE** *умова_відбору*]

елемент_вибірки

Вираз, або назва поля, значення якого потрібно вибрати. Символ «*» позначає всі поля.

перелік_таблиць

Назва таблиці, з якої здійснюється вибір значень.

умова_відбору

Вказує умови відбору потрібних записів.

DISTINCT | DISTINCTROW

Видалення з результату рядків-дублікатів. За замовчуванням вибираються всі рядки.

Для того, щоб виконати операцію об'єднання таблиць, потрібно за допомогою команди UNION об'єднати результати вибору рядків з двох, або більше, таблиць. Наведемо синтаксис команди.

SELECT ...

UNION [ALL | DISTINCT] SELECT ... [UNION [ALL | DISTINCT] SELECT ...]

Хід роботи.

Для початку я заповнив таблицю Car десятьма різними автомобілями

	id	mark	model	product_date	price	car_type
►	1	BMW	e34	1985-03-12	4500	Sedan
	2	Mercedes-Benz	g63	2019-07-31	10000	Sedan
	3	BMW	m5	2018-04-28	23500	Sedan
	4	Toyota	Corolla	1980-12-02	3750	Hatchback
	5	Mercedes-Benz	e300	2017-10-07	9800	Universal
	6	AUDI	a4	2016-04-03	5000	Universal
	7	AUDI	q7	2017-09-09	13000	er
	8	Toyota	Prado	2010-05-13	10000	Crossover
	9	Toyota	Tundra	2017-04-19	8500	Pickup
	10	AUDI	r8	2018-09-26	17000	Cabriolet
*	NULL	NULL	NULL	NULL	NULL	NULL

Потім я створив дві множини – дві вибірки з цієї таблиці – та зберіг їх у нових таблицях result1 та result2

```
CREATE TABLE IF NOT EXISTS result1 AS SELECT mark, model,
price, car_type
FROM car WHERE car_type != 'Sedan';
```

	mark	model	price	car_type
►	Toyota	Corolla	3750	Hatchback
	Mercedes-Benz	e300	9800	Universal
	AUDI	a4	5000	Universal
	AUDI	q7	13000	Crossover
	Toyota	Prado	10000	Crossover
	Toyota	Tundra	8500	Pickup
	AUDI	r8	17000	Cabriolet

```
CREATE TABLE IF NOT EXISTS result2 AS SELECT mark, model,
price, car_type
FROM car WHERE price >= 10000;
```

	mark	model	price	car_type
►	Mercedes-Benz	g63	10000	Sedan
	BMW	m5	23500	Sedan
	AUDI	q7	13000	Crossover
	Toyota	Prado	10000	Crossover
	AUDI	r8	17000	Cabriolet

1. Виконаємо об'єднання цих множин

```
SELECT * FROM result1 UNION SELECT * FROM result2;
```

	mark	model	price	car_type
►	Toyota	Corolla	3750	Hatchback
	Mercedes-Benz	e300	9800	Universal
	AUDI	a4	5000	Universal
	AUDI	q7	13000	Crossover
	Toyota	Prado	10000	Crossover
	Toyota	Tundra	8500	Pickup
	AUDI	r8	17000	Cabriolet
	Mercedes-Benz	g63	10000	Sedan
	BMW	m5	23500	Sedan

2. Виконаємо перетин двох множин

```
SELECT * FROM result1 WHERE mark IN
(SELECT mark FROM result2)
AND model IN (SELECT model FROM result2);
```

	mark	model	price	car_type
►	AUDI	q7	13000	Crossover
	Toyota	Prado	10000	Crossover
	AUDI	r8	17000	Cabriolet

3. Виконаємо різницю множин

```
SELECT * FROM result1 WHERE model NOT IN (SELECT model FROM
result2);
```

	mark	model	price	car_type
►	Toyota	Corolla	3750	Hatchback
	Mercedes-Benz	e300	9800	Universal
	AUDI	a4	5000	Universal
	Toyota	Tundra	8500	Pickup

4. Виконаємо операцію декартового множення множин

`SELECT * FROM result1,result2;`

	mark	model	price	car_type	mark	model	price	car_type
►	Toyota	Corolla	3750	Hatchback	Mercedes-Benz	g63	10000	Sedan
	Toyota	Corolla	3750	Hatchback	BMW	m5	23500	Sedan
	Toyota	Corolla	3750	Hatchback	AUDI	q7	13000	Crossover
	Toyota	Corolla	3750	Hatchback	Toyota	Prado	10000	Crossover
	Toyota	Corolla	3750	Hatchback	AUDI	r8	17000	Cabriolet
	Mercedes-Benz	e300	9800	Universal	Mercedes-Benz	g63	10000	Sedan
	Mercedes-Benz	e300	9800	Universal	BMW	m5	23500	Sedan
	Mercedes-Benz	e300	9800	Universal	AUDI	q7	13000	Crossover
	Mercedes-Benz	e300	9800	Universal	Toyota	Prado	10000	Crossover
	Mercedes-Benz	e300	9800	Universal	AUDI	r8	17000	Cabriolet
	AUDI	a4	5000	Universal	Mercedes-Benz	g63	10000	Sedan
	AUDI	a4	5000	Universal	BMW	m5	23500	Sedan
	AUDI	a4	5000	Universal	AUDI	q7	13000	Crossover
	AUDI	a4	5000	Universal	Toyota	Prado	10000	Crossover
	AUDI	a4	5000	Universal	AUDI	r8	17000	Cabriolet
	AUDI	q7	13000	Crossover	Mercedes-Benz	g63	10000	Sedan
	AUDI	q7	13000	Crossover	BMW	m5	23500	Sedan
	AUDI	q7	13000	Crossover	AUDI	q7	13000	Crossover
	AUDI	q7	13000	Crossover	Toyota	Prado	10000	Crossover
	AUDI	q7	13000	Crossover	AUDI	r8	17000	Cabriolet
	Toyota	Prado	10000	Crossover	Mercedes-Benz	g63	10000	Sedan
	Toyota	Prado	10000	Crossover	BMW	m5	23500	Sedan
	Toyota	Prado	10000	Crossover	AUDI	q7	13000	Crossover
	Toyota	Prado	10000	Crossover	Toyota	Prado	10000	Crossover
	Toyota	Prado	10000	Crossover	AUDI	r8	17000	Cabriolet
	Toyota	Tundra	8500	Pickup	Mercedes-Benz	g63	10000	Sedan
	Toyota	Tundra	8500	Pickup	BMW	m5	23500	Sedan
	Toyota	Tundra	8500	Pickup	AUDI	q7	13000	Crossover
	Toyota	Tundra	8500	Pickup	Toyota	Prado	10000	Crossover
	Toyota	Tundra	8500	Pickup	AUDI	r8	17000	Cabriolet
	AUDI	r8	17000	Cabriolet	Mercedes-Benz	g63	10000	Sedan
	AUDI	r8	17000	Cabriolet	BMW	m5	23500	Sedan
	AUDI	r8	17000	Cabriolet	AUDI	q7	13000	Crossover
	AUDI	r8	17000	Cabriolet	Toyota	Prado	10000	Crossover
	AUDI	r8	17000	Cabriolet	AUDI	r8	17000	Cabriolet

Висновок.

В ході лабораторної роботи я набув навичок роботи з даними в таблицях БД MySQL. Навчився виконувати операції перетину, об'єднання, різниці та декартового добутку.