

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра систем штучного інтелекту

Лабораторна робота №13
із дисципліни
Бази даних

Виконав:
Ст. групи КН-207
Букін С.А.
Прийняв:
Мельникова Н.І.

Львів – 2019 р.

Мета роботи: Навчитися аналізувати роботу СУБД та оптимізовувати виконання складних запитів на вибірку даних. Виконати аналіз складних запитів за допомогою директиви EXPLAIN, модифікувати найповільніші запити з метою їх пришвидчення.

Хід роботи.

1. Визначимо індекси таблиці Employee та Passport.
2. Створимо додаткові індекси для таблиць.
3. Дослідимо процес виконання запитів за допомогою EXPLAIN.

1. За допомогою директиви SHOW INDEX визначимо наявні індекси для таблиць Employee та Passport.

```
mysql> show index from employee;
```

Table	Non-unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
employee	0	PRIMARY	1	EmployeeID	A	12	NULL	NULL		BTREE			YES	NULL
employee	0	PRIMARY	2	RegionID	A	12	NULL	NULL		BTREE			YES	NULL
employee	0	PRIMARY	3	RoleID	A	12	NULL	NULL		BTREE			YES	NULL
employee	0	Login_UNIQUE	1	Login	A	12	NULL	NULL		BTREE			YES	NULL
employee	0	UQ_Employee_E-mail	1	E-mail	A	12	NULL	NULL		BTREE			YES	NULL
employee	1	RoleID_fk_idx	1	RoleID	A	4	NULL	NULL		BTREE			YES	NULL
employee	1	RegionID_fk1	1	RegionID	A	12	NULL	NULL		BTREE			YES	NULL

7 rows in set (0.62 sec)

```
mysql> show index from passport;
```

Table	Non-unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
passport	0	PRIMARY	1	EmployeeID	A	7	NULL	NULL		BTREE			YES	
passport	0	PRIMARY	2	RegionID	A	8	NULL	NULL		BTREE			YES	
passport	0	PRIMARY	3	Manufacturing_timeID	A	9	NULL	NULL		BTREE			YES	
passport	0	PRIMARY	4	PersonID	A	9	NULL	NULL		BTREE			YES	
passport	0	PRIMARY	5	StatusID	A	9	NULL	NULL		BTREE			YES	
passport	0	PRIMARY	6	TypeID	A	9	NULL	NULL		BTREE			YES	
passport	0	PRIMARY	7	ResonID	A	9	NULL	NULL		BTREE			YES	
passport	0	RecordID_UNIQUE	1	RecordID	A	9	NULL	NULL		BTREE			YES	
passport	0	PassportID_UNIQUE	1	PassportID	A	9	NULL	NULL		BTREE			YES	
passport	1	ResonID_fk_idx	1	ResonID	A	5	NULL	NULL		BTREE			YES	
passport	1	Manufacturing_timeID_fk_idx	1	Manufacturing_timeID	A	2	NULL	NULL		BTREE			YES	
passport	1	RegionID_fk_idx	1	RegionID	A	8	NULL	NULL		BTREE			YES	
passport	1	StatusID_fk_idx	1	StatusID	A	3	NULL	NULL		BTREE			YES	
passport	1	TypeID_fk	1	TypeID	A	3	NULL	NULL		BTREE			YES	
passport	1	PersonID_fk2	1	PersonID	A	7	NULL	NULL		BTREE			YES	

2. Створимо новий індекс для таблиць Employee та Passport.

```
mysql> create index employeeidx on employee (login, employeeid);
Query OK, 0 rows affected (0.83 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create index passportidx on passport (employeeid, date_of_registration);
Query OK, 0 rows affected (1.08 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

3. Виконаємо аналіз виконання складного запиту з однією з попередніх робіт використовуючи EXPLAIN та опцію STRAIGHT_JOIN.

```
mysql> explain select type_name, count(passport.passportid) as amount
-> From (employee inner join passport) inner join type
-> on employee.login = login
-> and employee.employeeid = passport.employeeid
-> and passport.typeid = type.typeid
-> where passport.date_of_registration between '2018-01-01' and '2019-06-01'
-> group by type_name;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	passport	NONE	ALL	PRIMARY,TypeID_fk,passportidx	NONE	NONE	NONE	9	11.11	Using where; Using temporary
1	SIMPLE	type	NONE	eq_ref	PRIMARY	PRIMARY	2	passportdb.passport.TypeID	1	100.00	NONE
1	SIMPLE	employee	NONE	ref	PRIMARY	PRIMARY	2	passportdb.passport.EmployeeID	1	100.00	NONE

3 rows in set, 1 warning (0.11 sec)

```
mysql> explain select straight_join type_name, count(passport.passportid) as amount
-> From (employee inner join passport) inner join type
-> on employee.login = login
-> and employee.employeeid = passport.employeeid
-> and passport.typeid = type.typeid
-> where passport.date_of_registration between '2018-01-01' and '2019-06-01'
-> group by type_name;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employee	NONE	index	PRIMARY	Login_UNIQUE	122	NONE	12	100.00	Using index; Using temporary
1	SIMPLE	passport	NONE	ALL	PRIMARY,TypeID_fk,passportidx	NONE	NONE	NONE	9	11.11	Using where; Using join buffer (Block Nested Loop)
1	SIMPLE	type	NONE	eq_ref	PRIMARY	PRIMARY	2	passportdb.passport.TypeID	1	100.00	NONE

3 rows in set, 1 warning (0.00 sec)

Висновок: На даній лабораторній роботі я навчився аналізувати і оптимізувати виконання запитів. Для аналізу запитів було використано директиву EXPLAIN, а для оптимізації – модифікація порядку з'єднання таблиць і створення додаткових індексів.