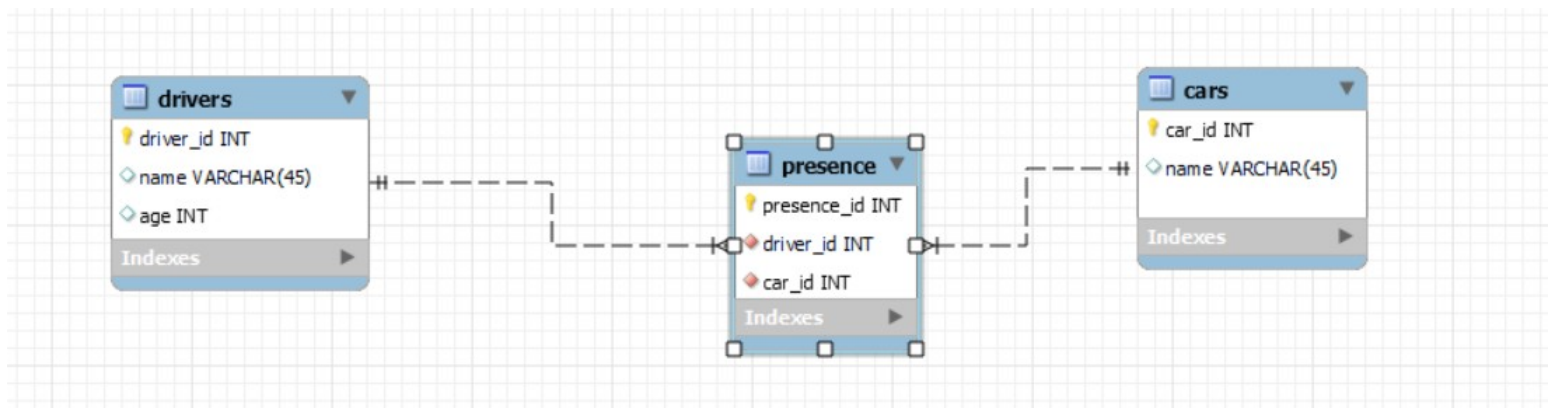


Задание

1. Установить отладчик и продемонстрировать выполнение процедуры по шагам
2. Сгенерировать большие (> 100000 записей) таблицы со случайными данными - числами и строками (с помощью хранимой процедуры)
Сделать 2 процедуры: с поэлементным добавлением записей и с транзакцией
3. Создать индексы (на отдельные поля и составные)
Сравнить время заполнения таблиц с индексами и без них
4. Сравнить время выполнения запросов (с условием отбора и сортировкой) с индексами и без

Создам вспомогательную новую базу данных



1 Установить отладчик и продемонстрировать выполнение процедуры по шагам

▶

```
set @a = 200;
set @b = 100;

set @c = @a - @b;

SELECT @c AS result;
```

●

```
set @a = 200;
set @b = 100;

set @c = @a - @b;

SELECT @c AS result;
```

result
100

Освоили режим отладки

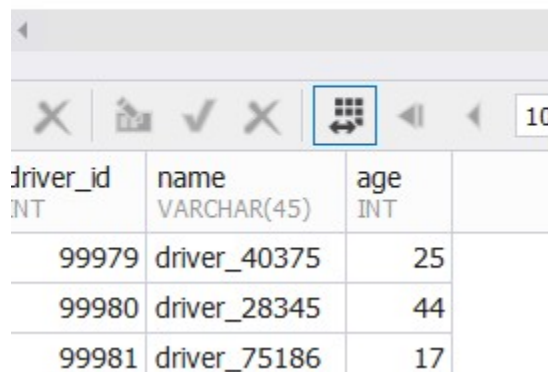
2 Сгенерировать большие (> 100000 записей) таблицы со случайными данными - числами и строками (с помощью хранимой процедуры). Сделать 2 процедуры: с поэлементным добавлением записей и с транзакцией

Заполню таблицу *drivers* (поэлементное добавление)

```
DROP PROCEDURE IF EXISTS driver_insert;
DELIMITER //
CREATE PROCEDURE driver_insert()
BEGIN
    DECLARE i INT DEFAULT 1;
    DECLARE rndName VARCHAR(50);
    DECLARE rndAge INT;
    WHILE i <= 100000 DO
        SET rndName = CONCAT('driver_', FLOOR(RAND() * 100000));
        SET rndAge = FLOOR(RAND() * 86);
        INSERT INTO drivers (name, age) VALUES (rndName, rndAge);
        SET i = i + 1;
    END WHILE;
END //
DELIMITER ;
```

В итоге

```
SELECT * FROM drivers;
```

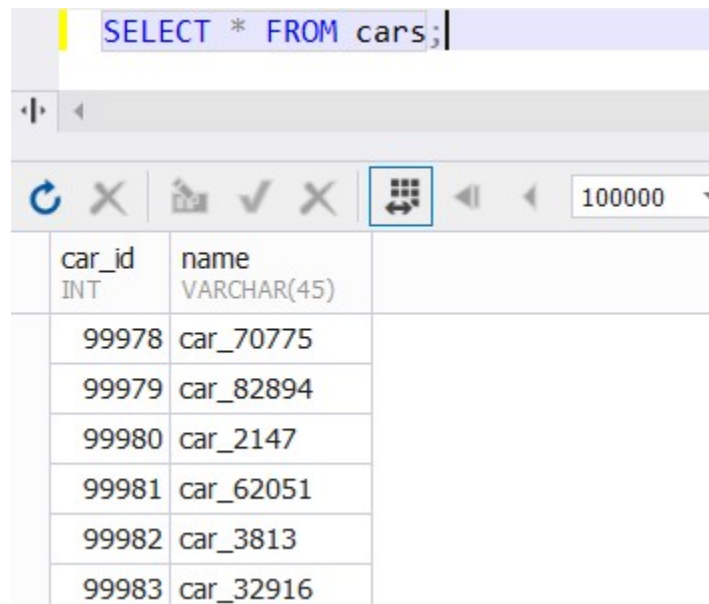


driver_id INT	name VARCHAR(45)	age INT
99979	driver_40375	25
99980	driver_28345	44
99981	driver_75186	17

Заполню таблицу *cars* (транзакция)

```
DROP PROCEDURE IF EXISTS car_insert;
DELIMITER //
CREATE PROCEDURE car_insert()
BEGIN
    DECLARE i INT DEFAULT 1;
    DECLARE rndName VARCHAR(50);
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        SELECT 'Error occurred, rolling back changes';
    END;
    START TRANSACTION;
    WHILE i <= 100000 DO
        SET rndName = CONCAT('car_', FLOOR(RAND() * 100000));
        INSERT INTO cars (name) VALUES (rndName);
        SET i = i + 1;
    END WHILE;
    COMMIT;
END //
DELIMITER ;
```

Итог



The screenshot shows a database client interface. At the top, a SQL query is entered in a text box: `SELECT * FROM cars;`. Below the query box is a toolbar with various icons, including a refresh icon (circular arrow) and a table icon (grid of squares). To the right of the toolbar is a dropdown menu showing the value `100000`. Below the toolbar, a table of results is displayed. The table has two columns: `car_id` (type `INT`) and `name` (type `VARCHAR(45)`). The table contains six rows of data, showing car IDs from 99978 to 99983 and their corresponding names.

car_id INT	name VARCHAR(45)
99978	car_70775
99979	car_82894
99980	car_2147
99981	car_62051
99982	car_3813
99983	car_32916

3 Создать индексы (на отдельные поля и составные). Сравнить время заполнения таблиц с индексами и без них

До создания индекса

Status	Query Text	Size (Bytes)	Executed On	Duration
✓	CALL car_insert()	42	01.05.2024 21:...	00:00:04.013

Создам простой индекс на поле

```
CREATE INDEX idx_name ON cars(name);
```

Заполнение после создания индекса

✓	#CREATE INDEX idx_...	122	01.05.2024 21:...	00:00:04.803
✗	CREATE INDEX idx_na, #CREATE INDEX idx_name ON cars(name);			0:00.018
✗	CREATE INDEX idx_na, CALL car_insert();			0:00.040

Видим, что получили прирост времени выполнения.

4 Сравнить время выполнения запросов (с условием отбора и сортировкой) с индексами и без

Сортировка

```
SELECT * FROM drivers order BY NAME DESC, age;
```


До создания индекса

Status	Query Text	Size (Bytes)	Executed On	Duration
✓	SELECT * FROM drive...	92	01.05.2024 21:...	00:00:00.233

Создам индекс

```
CREATE INDEX idx_sorted ON drivers (name DESC, age);
```

После создания


Status	Query Text	Size (Bytes)	Executed On ▼	Duration
	SELECT * FROM drive...	92	01.05.2024 21:...	00:00:00.179

Видим, что время уменьшилось

Условие отбора

```
SELECT name, age FROM drivers WHERE age BETWEEN 18 AND 41;
```


До создания индекса

Status	Query Text	Size (Bytes)	Executed On ▼	Duration
	SELECT name, age F...	116	01.05.2024 21:...	00:00:00.073

Создам индекс

```
CREATE INDEX idx_case ON drivers (name, age);
```

После создания

Status	Query Text	Size (Bytes)	Executed On ▼	Duration
	SELECT name, age F...	116	01.05.2024 21:...	00:00:00.033

Видим, что время уменьшилось