Отчет к ДЗ Проекции и материализованные представления

Задание:

1. **Создание таблицы:**
   - Создайте таблицу sales с полями:
   - id (UInt32) — уникальный идентификатор продажи
   - product_id (UInt32) — идентификатор продукта
   - quantity (UInt32) — количество проданных единиц
   - price (Float32) — цена за единицу
   - sale_date (DateTime) — дата продажи
   - Заполните таблицу тестовыми данными.
2. **Создание проекции:**
   - Создайте проекцию для таблицы sales, которая будет агрегировать данные по product_id и считать общую сумму продаж (количество и сумма по цене) за каждый продукт.
3. **Создание материализованного представления:**
   - Создайте материализованное представление sales_mv, которое будет автоматически обновляться при вставке новых данных в таблицу sales. Оно должно хранить общие продажи по продуктам с полями:
   - product_id
   - total_quantity
   - total_sales
4. **Запросы к данным:**
   - Напишите запрос, который извлекает данные из проекции sales_projection.
   - Напишите запрос, который извлекает данные из материализованного представления sales_mv.
5. **Сравнение производительности:**
   - Сравните время выполнения запроса к основной таблице sales с запросом к проекции sales_projection и материализованному представлению sales_mv. Обратите внимание на разницу в производительности.

1. **Создание таблицы и ее наполнение первыми 150000 строками**

```
CREATE TABLE otus.sales
(
    id Int32 comment 'уникальный идентификатор продажи',
    product_id Int32 comment 'идентификатор продукта',
    quantity Int32 comment 'количество проданных единиц',
    price Float32 comment 'цена за единицу',
    sale_date DateTime comment 'дата продажи'
)
ENGINE MergeTree
ORDER BY id;
```

```
INSERT INTO otus.sales
SELECT
    number AS id,
    rand() % 100 AS product_id,
    rand() % 10 AS quantity,
    rand() / 100.0 AS price,
    now() - INTERVAL rand() % 365 DAY AS sale_date
FROM system.numbers
LIMIT 150000;
```

2. **Создание проекции и ее материализация**

```
ALTER TABLE otus.sales ADD PROJECTION sales_projection
(
    SELECT
        product_id,
        sum(quantity) AS total_quantity,
        sum(quantity * price) AS total_sales
    GROUP BY product_id
);

ALTER TABLE otus.sales MATERIALIZE PROJECTION sales_projection;
```

3. **Создание материализованное представление sales_mv**

```
CREATE MATERIALIZED VIEW otus.sales_mv
ENGINE = SummingMergeTree()
ORDER BY product_id
AS
SELECT
    product_id,
    sum(quantity) AS total_quantity,
    sum(quantity * price) AS total_sales
FROM otus.sales
GROUP BY product_id;
```

4. **Добавление новых строк (1500 000) что бы стрегирить мат.вью и запросы к проекции и мат.вью**

```
INSERT INTO otus.sales
SELECT
    number AS id,
    rand() % 100 AS product_id,
    rand() % 10 AS quantity,
    rand() / 100.0 AS price,
    now() - INTERVAL rand() % 365 DAY AS sale_date
FROM system.numbers
LIMIT 1500000;
```

```
SELECT
    product_id,
    sum(quantity) AS total_quantity,
    sum(quantity * price) AS total_sales
FROM otus.sales
GROUP BY product_id

Query id: f534bf63-cc2b-45a0-a065-b28c318b9016

     ┌─product_id─┬─total_quantity─┬─────────total_sales─┐
  1. │          0 │              0 │                   0 │
  2. │         66 │          90612 │ 1952683343603.2266 │
  3. │         94 │          61272 │ 1322266449997.8281 │
  4. │         69 │         137412 │ 2953321824057.8384 │
  5. │         89 │         138771 │ 2973179540471.9756 │
 77. │         23 │          45273 │  971814503887.4883 │
 78. │         11 │          15066 │ 326165816228.33203 │
 79. │         61 │          15163 │  323791753932.5337 │
 80. │         33 │          45153 │  964656558945.8494 │
 81. │         58 │         122328 │ 2619009743493.8745 │
 82. │         38 │         119976 │  2550070823371.211 │
 83. │         86 │          89418 │ 1927716462695.8418 │
 84. │         74 │          60392 │ 1298762210793.3809 │
 85. │         81 │          15294 │  328393108578.7284 │
 86. │         77 │         106001 │ 2275713590782.1704 │
 87. │         96 │          90852 │ 1952766121197.4924 │
 88. │         10 │              0 │                   0 │
 89. │         22 │          30212 │  646507188426.0957 │
 90. │         13 │          45417 │  971515827619.2012 │
 91. │         17 │         105784 │  2253210013776.969 │
 92. │         39 │         136872 │  2927968863253.866 │
 93. │         59 │         135180 │ 2881757654728.3887 │
 94. │         32 │          30558 │  654556355618.6243 │
 95. │         60 │              0 │                   0 │
 96. │         76 │          91848 │ 1970025027853.9497 │
 97. │         80 │              0 │                   0 │
 98. │         75 │          76145 │      1626261393370 │
 99. │         87 │         106421 │ 2297844562430.2793 │
100. │         97 │         105770 │   2272229796344.92 │
     └─product_id─┴─total_quantity─┴─────────total_sales─┘

100 rows in set. Elapsed: 0.004 sec.
```

```
SELECT
    product_id,
    total_quantity,
    total_sales
FROM otus.sales_mv

Query id: 937673ef-8beb-414b-8095-159e66603dd1

    ┌─product_id─┬─total_quantity─┬──────────total_sales─┐
 1. │          1 │          15070 │  323318730239.1575 │
 2. │          2 │          30016 │  647434307725.3389 │
 3. │          3 │          45030 │  975212418359.2771 │
 4. │          4 │          59972 │ 1297831138393.0972 │
 5. │          5 │          75525 │  1616505067957.472 │
 6. │          6 │          90756 │ 1952808218099.4375 │
 7. │          7 │         104167 │ 2258345518717.8545 │
    │        ... │            ... │                ... │
83. │         92 │          30408 │  654305251605.9688 │
84. │         93 │          44574 │   963574747892.324 │
85. │         94 │          60664 │ 1307766968938.8125 │
86. │         95 │          75310 │  1632422671547.273 │
87. │         96 │          89964 │ 1932065771091.9612 │
88. │         97 │         104776 │   2250170390654.92 │
89. │         98 │         119600 │  2560942876855.375 │
90. │         99 │         133920 │ 2890517323311.2617 │
    └─product_id─┴─total_quantity─┴──────────total_sales─┘

90 rows in set. Elapsed: 0.002 sec.
```

При обращении к мат.вью запрос выполнился быстрее (0.002 против 0.004 сек). Возможно связано это с тем, что матвью отработала только с «вновь» добавленными строками.

Попробовал транкейтнуть таблицу sales и заново наполнил ее 15_000_000 строк. После селекта из матвью и проекции, проекция вновь отработала быстрее (0.003 против 0.005 сек).

Добавил в sales столбец

```
ALTER TABLE otus.sales ADD COLUMN datemodify DateTime default now();
```

На работе матвью и проекции это никак не сказалось. Запросы так же отработали.