

Лабораторная работа №4

Тема: Запросы на выборку данных по вариантам индивидуальных заданий

Цель работы

Усвоить способы создания представлений (views) средствами СУБД MS SQL Server на основе разработанной базы данных **Ugo1**. Закрепить навыки формирования виртуальных таблиц для удобства отчетности, просмотра и корректировки данных.

1. Индивидуальное задание

На основе базы данных **Ugo1** необходимо:

1. Создать представления для реализации отчетов в соответствии с требованиями индивидуального варианта.
 2. Создать представление по своему усмотрению на основании двух или более таблиц с целью просмотра и возможной корректировки базовых таблиц.
 3. Создать представление по своему усмотрению на основании одной таблицы с целью корректировки базовой таблицы.
-

2. Скрипты на создание представлений

2.1. Представление **vw_TotalObemByMarka**

Назначение: отчет по маркам угля с агрегированными показателями добычи (общий, средний, минимальный и максимальный объемы, количество рейсов).

```
CREATE VIEW vw_TotalObemByMarka AS
SELECT
    dv.MarkaUglia,
    u.Zolnost,
    u.Vlazhnost,
```

```

    u.TeplotaSgoraniya,
    SUM(dv.Obem) AS TotalObem,
    AVG(dv.Obem) AS AvgObem,
    MIN(dv.Obem) AS MinObem,
    MAX(dv.Obem) AS MaxObem,
    COUNT(*) AS KolichествоReisov
FROM DobychaVyvoz dv
JOIN Ugol u ON dv.MarkaUgla = u.MarkaUgla
GROUP BY dv.MarkaUgla, u.Zolnost, u.Vlazhnost, u.TeplotaSgoraniya;

```

Пример использования:

```
SELECT * FROM vw_TotalObemByMarka;
```

2.2. Представление **vw_ReisyFullInfo**

Назначение: сводный отчет по рейсам с информацией о марке угля, экскаваторе, самосвале, экскаваторщике и водителе самосвала.

```

CREATE VIEW vw_ReisyFullInfo AS
SELECT
    dv.NomerReisa,
    dv.Data,
    dv.Smena,
    dv.Obem,
    dv.MarkaUgla,
    u.Zolnost,
    u.Vlazhnost,
    u.TeplotaSgoraniya,
    e.KodEkskavatora,
    e.Nazvanie AS NazvanieEkskavatora,
    e.ObemKovsha,
    de.TabNomer AS TabNomerEkskavatorshik,
    r1.FIO AS FIOEkskavatorshik,
    s.KodSamosvala,
    s.Nazvanie AS NazvanieSamosvala,
    s.Tonnazh,
    vs.TabNomer AS TabNomerVoditel,
    r2.FIO AS FIOVoditel
FROM DobychaVyvoz dv
LEFT JOIN Ugol u ON dv.MarkaUgla = u.MarkaUgla
LEFT JOIN DobychaEkskavator de ON dv.NomerReisa = de.NomerReisa

```

```
LEFT JOIN Ekskavatory e ON de.KodEkskavatora = e.KodEkskavatora
LEFT JOIN Rabotniki r1 ON de.TabNomer = r1.TabNomer
LEFT JOIN VyvozSamosval vs ON dv.NomerReisa = vs.NomerReisa
LEFT JOIN Samosvaly s ON vs.KodSamosvala = s.KodSamosvala
LEFT JOIN Rabotniki r2 ON vs.TabNomer = r2.TabNomer;
```

Пример использования:

```
SELECT * FROM vw_ReisyFullInfo;
```

2.3. Представление **vw_RabotnikiVoditeli**

Назначение: выборка работников с должностью «Водитель» для удобства просмотра и корректировки их данных.

Представление основано на одной таблице **Rabotniki** и допускает обновление записей, попадающих в условие.

```
CREATE VIEW vw_RabotnikiVoditeli AS
SELECT
    TabNomer,
    FIO,
    IIN,
    SIK,
    KodDolzhnosti,
    KodFonda
FROM Rabotniki
WHERE KodDolzhnosti = 103
WITH CHECK OPTION;
```

Примеры использования:

-- Просмотр всех водителей

```
SELECT * FROM vw_RabotnikiVoditeli;
```

-- Обновление данных водителя через представление

```
UPDATE vw_RabotnikiVoditeli
```

```
SET SIK = 'S011'
```

```
WHERE TabNomer = 2;
```

3. Ответы на контрольные вопросы

1. Назначение механизма представлений

Механизм представлений позволяет создавать виртуальные таблицы на основе запросов `SELECT`. Представления используются для:

- упрощения работы с сложными структурами баз данных;
 - организации удобных форм отчетности;
 - разграничения доступа к данным (скрытие лишних колонок и строк);
 - обеспечения дополнительного уровня безопасности;
 - работы с агрегированными данными без изменения структуры базовых таблиц;
 - логического отделения пользовательского уровня от физической структуры данных.
-

2. Классификация представлений

Представления можно классифицировать по разным признакам.

1. По составу данных:

- представления с подмножеством колонок;
- представления с подмножеством строк;
- представления, основанные на соединении (`JOIN`) нескольких таблиц;
- агрегированные представления (с `SUM`, `AVG`, `COUNT`, `MAX`, `MIN` и др.).

2. По возможности модификации данных:

- обновляемые представления (обычно основаны на одной таблице, без агрегатных функций, `DISTINCT`, `UNION` и т.п.);
- не обновляемые представления (сложные соединения, агрегаты, объединения).

3. По назначению:

- отчетные представления;
 - представления для обеспечения безопасности и разграничения доступа;
 - представления для упрощения сложных запросов и логики приложений;
 - представления для работы с секционированными данными (объединение нескольких таблиц в одну логическую).
-

3. Способы создания представлений

В MS SQL Server представления создаются и изменяются:

1. С помощью T-SQL:

- **CREATE VIEW** — создание представления;
- **ALTER VIEW** — изменение существующего представления;
- **DROP VIEW** — удаление представления;

2. Дополнительно могут использоваться опции:

- **WITH ENCRYPTION** — шифрование определения представления;
- **WITH CHECK OPTION** — запрет модификаций, выводящих строку за пределы условия представления.

3. С помощью SQL Server Management Studio (SSMS):

- через папку **Views** → **New View** для визуального конструктора;
- **Design** для изменения структуры представления;
- возможность визуально добавлять таблицы, поля, условия, сортировку, группировку.

