

МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Ярославский государственный университет им. П.Г. Демидова»**

Отчет по лабораторной работе на тему

Музыкальный магазин

Дисциплина «Объектные базы данных»

Направление подготовки 02.04.02 Фундаментальная информатика и
информационные технологии

Студент группы ИТ-21МО

_____ Д.В. Грушевская

«28» ноября 2020 г.

Ярославль 2020 г.

Содержание

Задание	2
Описание БД «Музыкальный магазин»	3

Задание

ВАРИАНТ 2 (Грушевская Дарья)

Предметная область: «Музыкальный магазин».

Объекты:

- запись (идентификатор, название, время звучания, стиль);
- альбом (идентификатор, название, стоимость, количество на складе, количество проданных экземпляров);
- исполнитель (имя, псевдоним или название группы; страна).

Для каждого альбома должен быть указан список треков с указанием порядкового номера для каждой записи; одна запись может входить в разные альбомы; один трек также может быть записан сразу несколькими исполнителями. В каждом альбоме не более 30 записей.

Методы:

Минимальный функционал:

- 1) добавить запись (изначально указывается один исполнитель);
- 2) добавить исполнителя для записи (если указанная запись не добавлена ни в один альбом);
- 3) добавить исполнителя;
- 4) добавить альбом (изначально указывается один трек или ни одного);
- 5) добавить трек в альбом (если не продано ни одного экземпляра);
- 6) список альбомов в продаже (количество на складе больше 0);
- 7) список исполнителей;
- 8) поставка альбома (количество на складе увеличивается на указанное значение);
- 9) продать альбом (количество на складе уменьшается, проданных – увеличивается; продать можно только альбомы, в которых есть хотя бы один трек).
- 10) удалить исполнителей, у которых нет ни одной записи.

Основной функционал:

- 11) трек-лист указанного альбома с указанием суммарного времени звучания альбома;
- 12) выручка магазина (суммарная стоимость проданных альбомов по каждому в отдельности и по магазину в целом);
- 13) удалить трек с указанным номером из альбома с пересчётом остальных номеров (если не продано ни одного экземпляра альбома);
- 14) удалить исполнителя из записи (если запись не входит ни в один альбом и если этот исполнитель не единственный);
- 15) определить предпочитаемый музыкальный стиль указанного исполнителя (стиль, в котором записано большинство его треков).
- 16) определить предпочитаемый музыкальный стиль по каждой стране происхождения исполнителей;
- 17) определить авторство альбомов (для каждого альбома выводится исполнитель или список исполнителей, если все треки этого альбома записаны одним множеством исполнителей; в противном случае выводится «Коллективный сборник»).

Описание БД «Музыкальный магазин»

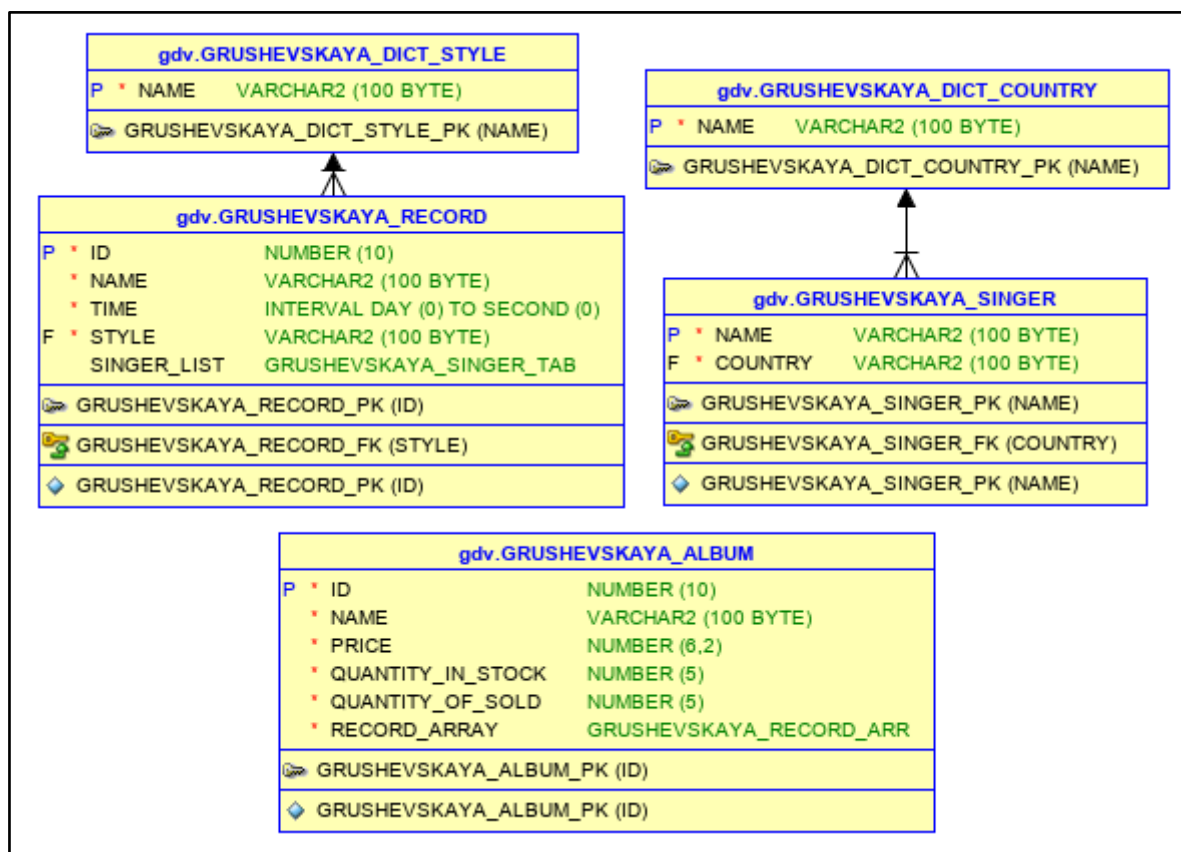


Рис. 1 Схема БД

Репозиторий: https://github.com/ggeraldina/sql_2020

Структура БД.

Помимо таблиц, требующихся по заданию, были созданы дополнительно 2 таблицы – словарь стран GRUSHEVSKAYA_DICT_COUNTRY и словарь стилей GRUSHEVSKAYA_DICT_STYLE. Данные таблицы помогают отслеживать уникальность стран и стилей. В результате не будет ситуации, когда один исполнитель из РФ, а другой из России, хотя это по сути одно и то же.

```

CREATE TABLE GRUSHEVSKAYA_DICT_COUNTRY(
    -- название страны
    NAME VARCHAR2(100 BYTE)
    PRIMARY KEY
    NOT NULL
);
    
```

```

CREATE TABLE GRUSHEVSKAYA_DICT_STYLE(
    -- название стиля
    NAME VARCHAR2(100 BYTE)
    PRIMARY KEY
    NOT NULL
);
    
```

В таблице GRUSHEVSKAYA_SINGER содержится информация об исполнителях (имя, псевдоним или название группы; страна из GRUSHEVSKAYA_DICT_COUNTRY).

```
CREATE TABLE GRUSHEVSKAYA_SINGER(  
  -- имя, псевдоним или название группы  
  NAME VARCHAR2(100 BYTE),  
  -- страна  
  COUNTRY VARCHAR2(100 BYTE)  
);
```

В таблице GRUSHEVSKAYA_RECORD содержится информация о записях (идентификатор; название; время звучания; стиль из таблицы GRUSHEVSKAYA_DICT_STYLE; список исполнителей). Трек может быть написан одним исполнителем или несколькими. Информация об исполнителях хранится в поле исполнители (SINGER_LIST), представляющим собой вложенную таблицу типа GRUSHEVSKAYA_SINGER_TAB. GRUSHEVSKAYA_SINGER_TAB – тип коллекции, таблица из имен исполнителей.

```
CREATE TYPE GRUSHEVSKAYA_SINGER_TAB AS TABLE OF VARCHAR2(100 BYTE);  
/  
CREATE TABLE GRUSHEVSKAYA_RECORD(  
  -- идентификатор  
  ID NUMBER(10,0),  
  -- название  
  NAME VARCHAR2(100 BYTE),  
  -- время звучания  
  TIME INTERVAL DAY (0) TO SECOND (0),  
  -- стиль  
  STYLE VARCHAR2(100 BYTE),  
  -- список исполнителей  
  SINGER_LIST GRUSHEVSKAYA_SINGER_TAB  
) NESTED TABLE SINGER_LIST  
  STORE AS GRUSHEVSKAYA_SINGER_LIST;
```

В таблице GRUSHEVSKAYA_ALBUM содержится информация об альбомах (идентификатор; название; стоимость; количество на складе; количество проданных экземпляров; список записей). Информация о записях, входящих в альбом, хранится в поле RECORD_ARRAY типа GRUSHEVSKAYA_RECORD_ARR. GRUSHEVSKAYA_RECORD_ARR – массив длины 30, состоящий из идентификаторов записей. Для хранения был выбран массив, так как в нем треки автоматически получают порядковый номер звучания.

```
CREATE TYPE GRUSHEVSKAYA_RECORD_ARR AS VARRAY(30) OF NUMBER(10,0);  
/  
CREATE TABLE GRUSHEVSKAYA_ALBUM(  
  -- идентификатор  
  ID NUMBER(10, 0),  
  -- название  
  NAME VARCHAR2(100 BYTE),
```

```

-- стоимость
PRICE NUMBER(6, 2),
-- количество на складе
QUANTITY_IN_STOCK NUMBER(5, 0),
-- количество проданных экземпляров
QUANTITY_OF_SOLD NUMBER(5, 0),
-- список (массив) записей
RECORD_ARRAY GRUSHEVSKAYA_RECORD_ARR
);

```

Все поля в таблицах являются обязательными (NOT NULL) за исключением SINGER_LIST, так как на вложенные таблицы нельзя наложить данное ограничение. Поэтому данное ограничение было реализовано с помощью триггера GRUSHEVSKAYA_TR_ON_RECORDS. Об устройстве триггера будет написано чуть позднее.

На поля таблиц были наложены естественные ограничения. С помощью CHECK: цена альбома, количество альбомов на складе, количество проданных альбомов не могут быть меньше нуля. Другие ограничения были связаны с полями типа вложенная таблица или тапа массив, поэтому ограничения были реализованы с помощью триггеров.

Внешние (F) и первичные (P) ключи изображены на схеме (см. рис 1). Помимо изображенных на схеме связей дополнительно была реализована связь «многие-ко-многим» таблиц SINGER-RECORD и RECORD-ALBUM. Эти связи были реализованы с помощью триггеров.

В БД было создано всего 6 триггеров:

- GRUSHEVSKAYA_TR_ON_RECORDS,
- GRUSHEVSKAYA_TR_ON_SINGERS_DEL,
- GRUSHEVSKAYA_TR_ON_SINGERS_UDP,
- GRUSHEVSKAYA_TR_ON_ALBUM,
- GRUSHEVSKAYA_TR_ON_RECORD_DEL,
- GRUSHEVSKAYA_TR_ON_RECORD_UDP.

GRUSHEVSKAYA_TR_ON_RECORDS – триггер уровня записи для таблицы GRUSHEVSKAYA_RECORD, срабатывающий до вставки или обновления строки.

1) Проверяет поле SINGER_LIST на NULL – имитирует работу ограничения NOT NULL для поля типа вложенная таблица.

2) Удаляет NULL значения из SINGER_LIST.

3) Проверяет на пустоту SINGER_LIST. Список исполнителей не должен быть пуст. Хотя бы один исполнитель должен быть.

4) В случае обновления исполнителей записи проверяет: не содержится ли данная запись в одном из альбомов. Исполнителей нельзя добавлять, удалять, если запись присутствует в одном из альбомов.

5) Частично имитирует работу внешнего ключа для SINGER-RECORD. Если подмножество исполнителей не соответствует таблице исполнителей, то отменяет вставку или "откатывает" обновление до прежней версии полей.

GRUSHEVSKAYA_TR_ON_SINGERS_DEL – триггер уровня записи для таблицы GRUSHEVSKAYA_SINGER, срабатывающий до удаления строки.

1) Частично имитирует работу внешнего ключа для SINGER-RECORD. Перед удалением исполнителя проверяет: нет ли у исполнителя записей. Если есть, то удалять исполнителя нельзя.

GRUSHEVSKAYA_TR_ON_SINGERS_UDP – составной триггер уровня команды для таблицы GRUSHEVSKAYA_SINGER, срабатывающий при обновлении поля NAME.

1) Частично имитирует работу внешнего ключа для SINGER-RECORD. После обновления имени исполнителя обновляет его имя для всех записей:

- В блоке AFTER уровня записи сохраняет значения изменяющихся имен в ассоциативном массиве, определяемом как глобальная переменная составного триггера.
- В блоке AFTER уровня команды перебираются все списки исполнителей в треках и вносятся все необходимые изменения с помощью сохранённого ассоциативного массива и булевого флага.

GRUSHEVSKAYA_TR_ON_ALBUM – триггер уровня записи для таблицы GRUSHEVSKAYA_ALBUM, срабатывающий до вставки или обновления строки.

1) Удаляет дубликаты из массива с помощью ассоциативного массива.

2) Проверяет был ли продан альбом. Если да, то обновлять, добавлять, удалять новые треки нельзя. Случаи на равенство старых, новых значений NULL-ю пришлось разграничить, так как возникало исключение.

3) Частично имитирует работу внешнего ключа для RECORD-ALBUM. Если подмножество записей не соответствует таблице записей, то отменяет вставку или "откатывает" обновление до прежней версии полей.

GRUSHEVSKAYA_TR_ON_RECORD_DEL – триггер уровня записи для таблицы GRUSHEVSKAYA_RECORD, срабатывающий до удаления строки.

1) Частично имитирует работу внешнего ключа для RECORD-ALBUM. Перед удалением трека проверяет: нет ли его в одном из альбомов. Если есть, то удалять запись нельзя.

GRUSHEVSKAYA_TR_ON_RECORD_UDP – составной триггер уровня команды для таблицы GRUSHEVSKAYA_RECORD, срабатывающий при обновлении поля ID.

1) Частично имитирует работу внешнего ключа для RECORD-ALBUM. После обновления идентификатора записи обновляет его идентификатор для всех альбомов:

- В блоке AFTER уровня записи сохраняет значения изменяющихся идентификаторов в ассоциативном массиве, определяемом как глобальная переменная составного триггера.
- В блоке AFTER уровня команды перебираются все списки записей в альбомах и вносятся все необходимые изменения с помощью сохранённого ассоциативного массива и булевого флага.

Все триггеры при нарушении какого-либо из условий выбрасывают исключения. Исключения перечислены в отдельном пакете GRUSHEVSKAYA_EXCEPTIONS.

Для того чтобы не вводить руками ID записей и альбомов, были созданы две последовательности (SEQUENCE). GRUSHEVSKAYA_NUM_RECORD и GRUSHEVSKAYA_NUM_ALBUM соответственно.

Методы.

Все методы реализованы в пакете GRUSHEVSKAYA_PACKAGE.

Методы и параметры прокомментированы. Что есть что можно понять из комментариев. Все методы являются процедурами и выводят сообщения на экран. Ниже приведена соответствующая часть скрипта.

Дополнительно были реализованы общедоступные методы добавления стран и стилей, а также приватный метод PRINT_MSG_EX, печатающий код и сообщение исключения. В идеале вызываться не должен.

-- Пакет GRUSHEVSKAYA_PACKAGE с реализованным функционалом

```
CREATE OR REPLACE
PACKAGE GRUSHEVSKAYA_PACKAGE AS
  -- Добавить страну в словарь.
  PROCEDURE ADD_IN_DICT_COUNTRY (
    -- Название страны
    NAME VARCHAR2
  );
  -- Добавить стиль в словарь.
  PROCEDURE ADD_IN_DICT_STYLE (
    -- Название стиля
    NAME VARCHAR2
  );

  -- Минимальный функционал

  -- 1) Добавить запись (изначально указывается один исполнитель).
  PROCEDURE ADD_RECORD (
    -- Название
    NAME VARCHAR2,
    -- Количество часов звучания
    HOURS NUMBER,
    -- Количество минут звучания
    MINUTES NUMBER,
    -- Количество секунд звучания
    SECONDS NUMBER,
    -- Стиль из словаря
    STYLE VARCHAR2,
    -- Имя исполнителя
    SINGER VARCHAR2
  );
  -- 2) Добавить исполнителя для записи
  -- (если указанная запись не добавлена ни в один альбом
  -- - Условие проверяется на уровне триггера).
```

```

PROCEDURE ADD_SINGER_IN_RECORD (
    -- ID записи
    RECORD_ID NUMBER,
    -- Имя исполнителя
    SINGER_NAME VARCHAR2
);
-- 3) Добавить исполнителя.
PROCEDURE ADD_SINGER (
    -- Имя (ФИО)
    NAME VARCHAR2,
    -- Страна из словаря
    COUNTRY VARCHAR2
);
-- 4) Добавить альбом (изначально указывается один трек или ни одного).
-- Реализация для добавления альбома с одной записью.
PROCEDURE ADD_ALBUM (
    -- Название
    NAME VARCHAR2,
    -- Цена (>= 0)
    PRICE NUMBER,
    -- Количество на складе (>= 0)
    QUANTITY_IN_STOCK NUMBER,
    -- Количество проданных альбомов (>= 0)
    QUANTITY_OF_SOLD NUMBER,
    -- ID добавляемой записи
    RECORD_ID NUMBER
);
-- 4) Добавить альбом (изначально указывается один трек или ни одного).
-- Реализация для добавления альбома без записей.
PROCEDURE ADD_ALBUM (
    -- Название
    NAME VARCHAR2,
    -- Цена (>= 0)
    PRICE NUMBER,
    -- Количество на складе (>= 0)
    QUANTITY_IN_STOCK NUMBER,
    -- Количество проданных альбомов (>= 0)
    QUANTITY_OF_SOLD NUMBER
);
-- 5) Добавить трек в альбом
-- (если не продано ни одного экземпляра
-- - Условие проверяется на уровне триггера).
PROCEDURE ADD_RECORD_IN_ALBUM (
    -- ID альбома
    ALBUM_ID NUMBER,
    -- ID добавляемой записи
    RECORD_ID NUMBER
);

```



```

-- 6) Список альбомов в продаже (количество на складе больше 0).
PROCEDURE PRINT_ALBUMS_IN_STOCK;

-- 7) Список исполнителей.
PROCEDURE PRINT_SINGERS;

-- 8) Поставка альбома
-- (количество на складе увеличивается на указанное значение).
PROCEDURE ADD_ALBUMS_IN_STOCK (
    -- ID альбома
    ALBUM_ID NUMBER,
    -- Количество
    QUANTITY NUMBER
);

-- 9) Продать альбом
-- (количество на складе уменьшается, проданных – увеличивается;
-- продать можно только альбомы, в которых есть хотя бы один трек
-- - Условие проверяется в самой функции).
PROCEDURE SELL_ALBUMS(
    -- ID альбома
    ALBUM_ID NUMBER,
    -- Количество
    QUANTITY NUMBER
);

-- 10) Удалить исполнителей, у которых нет ни одной записи.
PROCEDURE DELETE_SINGERS_WITHOUT_RECORDS;

-- Основной функционал

-- 11) Трек-лист указанного альбома
-- с указанием суммарного времени звучания альбома.
PROCEDURE PRINT_ALBUM_RECORDS(ALBUM_ID NUMBER);

-- 12) Выручка магазина
-- (суммарная стоимость проданных альбомов
-- по каждому в отдельности
-- и по магазину в целом).
PROCEDURE PRINT_INCOME;

-- 13) Удалить трек с указанным номером из альбома
-- с пересчётом остальных номеров
-- (если не продано ни одного экземпляра альбома
-- - Условие проверяется на уровне триггера).
PROCEDURE DELETE_RECORD_FROM_ALBUM(
    -- ID альбома
    ALBUM_ID NUMBER,
    -- Номер звучания записи в альбоме
    RECORD_NUMBER NUMBER
);

-- 14) Удалить исполнителя из записи
-- (если запись не входит ни в один альбом
-- и если этот исполнитель не единственный

```

```

-- - Условия проверяются на уровне триггера).
PROCEDURE DELETE_SINGER_FROM_RECORD(
  -- ID записи
  RECORD_ID NUMBER,
  -- Номер исполнителя в списке
  SINGER_NUMBER NUMBER
);
-- 15) Определить предпочитаемый музыкальный стиль указанного исполнителя
-- (стиль, в котором записано большинство его треков).
PROCEDURE PRINT_SINGER_STYLE(
  -- Имя исполнителя
  SINGER_NAME VARCHAR2
);
-- 16) Определить предпочитаемый музыкальный стиль
-- по каждой стране происхождения исполнителей.
PROCEDURE PRINT_COUNTRY_STYLE;
-- 17) Определить авторство альбомов
-- (для каждого альбома выводится
-- исполнитель или список исполнителей,
-- если все треки этого альбома записаны
-- одним множеством исполнителей;
-- в противном случае выводится «Коллективный сборник»).
PROCEDURE PRINT_ALBUM_AUTHOR(
  -- ID альбома
  ALBUM_ID NUMBER
);
END;
```