# Задание № 1

# Создание и заполнение базы данных в СУБД Oracle

Целью настоящего задания является изучение возможности создания баз данных посредством инструкций SQL, правила задания ограничений различных типов, а также запросов на добавление, обновление, удаление информации из базы данных

**Теоретические сведения**

Единицей работы для реляционных баз данных является таблица. Инструкция *create table* SQL позволяет создавать таблицы БД и ограничения на значения столбцов, а также создавать связи типа первичный ключ - внешний ключ между таблицами.

Синтаксическая диаграмма данной инструкции представлена на рисунке 1.





Рис. 1

Продемонстрируем использование инструкции *create table* на примере создания таблицы ORDERS. Так, на рисунке 2 продемонстрированы таблицы и предполагаемые связи, создаваемые между ними.



Рис. 2

Инструкция *create table* для создания таблицы ORDERS, при условии существования трех её родительских таблиц в базе данных, выглядит следующим образом:

*create table* orders (ord\_n *integer not null*,

ord\_date *date not null*,

cust\_n *varchar2*(5) *not null*,

sale\_n *varchar2*(5) *not null*,

prod\_n *varchar2*(10) *not null*,

qty *integer*,

amount *integer*,

*primary key* (ord\_n),

*unique* (ord\_n),

*constraint* placed\_by *foreign key* (cust\_n)

*references* customers *on delete cascade*,

*constraint* taken\_by *foreign key* (sale\_n) *references* salesreps,

*constraint* is\_for *foreign key* (prod\_n) *references* products);

В вышеприведенном примере при задании “внешнего ключа” используется опция *on delete cascade,* реализующая правило “каскадного удаления” строк из таблицы при удалении соответствующих “строк - предков” из “родительской ” таблицы. По умолчанию удаление строк, у которых есть потомки – запрещено.

Во многих коммерческих СУБД при создании базы данных для некоторых столбцов таблиц можно назначить условия на принимаемые ими значения. Данные условия задаются в инструкциях по созданию таблиц посредством ограничения *check* . В качестве примера можно рассмотреть фрагмент инструкции *create table* для создания таблицы staff (сотрудники):

*create table* staff (sno *integer not null*,

age *integer*,

*check* (sno *between* 101 *and* 199),

*check* (age >=21));

Здесь столбец age определяет возраст сотрудников. В соответствии с ограничением *check* значение этого столбца не могут быть меньше 21. Точно также значения для столбца sno ограничены диапазоном [101, 199].

Для выполнения задания требуется использование инструкции добавления данных в таблицы, а также возможно инструкции по изменению и удалению данных из таблиц. Рассмотрим их применение подробнее.

ДОБАВЛЕНИЕ НОВЫХ ДАННЫХ

Наименьшей единицей информации, которую можно добавить в реляционную базу данных, является одна строка. В реляционной СУБД существует три способа добавления новых строк в базу данных:

• Однострочная инструкция INSERT позволяет добавить в таблицу одну новую строку. Она широко используется в повседневных приложениях, например в программах ввода данных.

• Многострочная инструкция INSERT обеспечивает извлечение строк из одной части базы данных и добавление их в другую таблицу. Она обычно используется, когда “старые” строки таблицы пересылаются в неиспользуемую таблицу для сохранения.

Однострочная инструкция INSERT, добавляет в таблицу новую строку. Синтаксическая диаграмма этой инструкции представлена на рисунке 10.



Рис. 3

В предложении INTO указывается таблица (*целевая таблица*), в которую добавляется новая строка, а в предложении VALUES содержатся значения данных для новой строки. Список столбцов определяет, какие значения, в какой столбец заносятся.

На практике данные почти всегда добавляются в базу с помощью программ ввода данных, использующих специальные формы. После окончания ввода данных в форму, приложение добавляет новую строку данных в базу с помощью программного SQL. Однако инструкцию INSERT можно использовать и в интерактивном режиме. Независимо от того, какой SQL используется, интерактивный или программный, инструкция INSERT имеет один и тот же вид.

Список столбцов в инструкции INSERT служит для того, чтобы установить соответствие между значениями данных, содержащимися в предложении VALUES, и столбцами, для которых эти данные предназначены. Списки значений и столбцов должны содержать одинаковое число элементов, а тип данных каждого значения должен быть совместимым с типом соответствующего столбца, иначе произойдет ошибка.

При добавлении в таблицу новой строки всем столбцам, имена которых отсутствуют в списке столбцов инструкции INSERT, автоматически присваивается значение NULL. Значение NULL можно присвоить и явным образом, включив эти столбцы в список, а в списке значений задав для них ключевое слово NULL. Для удобства в SQL разрешается не включать список столбцов в инструкцию INSERT. Если список столбцов опущен, он генерируется автоматически и в нем слева направо перечисляются все столбцы таблицы. При выполнении инструкции SELECT \* генерируется такой же список столбцов. Если список столбцов опущен, то в списке значений необходимо явно указывать значения NULL. Кроме того, последовательность значений данных должна в точности соответствовать порядку столбцов в таблице.

Многострочная инструкция INSERT, синтаксическая диаграмма которой изображена на рисунке 4, добавляет в целевую таблицу несколько строк (более одной).



Рис. 4

В этой разновидности инструкции INSERT значения для новых строк явно не задаются. Источником новых строк служит запрос на выборку, содержащийся внутри инструкции.

УДАЛЕНИЕ СУЩЕСТВУЮЩИХ ДАННЫХ

Инструкция DELETE, синтаксическая диаграмма которой изображена на рисунке 5, удаляет выбранные строки из одной таблицы.



Рис. 5

В предложении FROM указывается таблица, содержащая строки, которые требуется удалить. В предложении WHERE указывается критерий отбора строк, которые должны быть удалены.

ОБНОВЛЕНИЕ СУЩЕСТВУЮЩИХ ДАННЫХ

Инструкция UPDATE, синтаксическая диаграмма которой изображена на рисунке 6, обновляет значения одного или нескольких столбцов в выбранных строках одной таблицы.



Рис 6

В инструкции указывается целевая таблица, которая должна быть модифицирована, при этом пользователь должен иметь разрешение на обновление таблицы и каждого конкретного столбца. Предложение WHERE отбирает строки таблицы, подлежащие обновлению. В предложении SET указывается, какие столбцы должны быть обновлены, и для них задаются новые значения.

**Варианты заданий**

**1.** Создайте БД отдела кадров университета. Ориентировочные таблицы-составляющие: «Сотрудники», **«Контракты»**, «Кафедры», «Должности».

**2.** Создайте БД галантерейного магазина. Ориентировочные таблицы-составляющие: «Партии товара», «Поставщики», **«Продажи**», «Продавцы».

**3.** Создайте БД отдела доставки почтового отделения. Ориентировочные таблицы-составляющие: «Подписчики», «Периодика», **«Подписки»**, «Почтальоны».

**4.** Создайте БД универмага. Ориентировочные таблицы-составляющие: «Товары», «Отделы», **«Продажа»**, «Продавцы».

**5.** Создайте БД штатного расписания предприятия. Ориентировочные таблицы-составляющие: «Отделы», «Должности», «Тип производства», «Сотрудники», **«Штатное расписание»**.

**6.** Создайте БД фирмы по производству пиломатериалов. Ориентировочные таблицы-составляющие: «Изделия», «Сырье», **«Продажи»**, «Поставщики».

**7.** Создайте БД программы выпуска деталей литейного цеха. Ориентировочные таблицы-составляющие: «Детали», «Материал», **«Технологические процессы»**, «Рабочие».

**8.** Создайте БД инфекционного отделения городской больницы. Ориентировочные таблицы-составляющие: «Койко-место», «Больные», «Диагнозы», **«Госпитализация»**, «Врачи».

**9.** Создайте БД процессов обработки партий деталей. Ориентировочные таблицы-составляющие: «Детали», «Оборудование», **«Технологические карты»**, «Процессы», «Бригады».

**10.** Создайте БД фирмы по оптовой реализации бытовой техники. Ориентировочные таблицы-составляющие: «Продукция», «Клиенты», **«Заказы»**, «Производители».

**11.** Создайте БД оборудования НИИ. Ориентировочные таблицы-составляющие: «Оборудование», «Исследовательские работы», **«Акты проведения работ»**, «Руководители работ».

**12.** Создайте БД фирмы по производству столярных работ. Ориентировочные таблицы-составляющие: «Продукция», «Клиенты», **«Договора»**, «Услуги», «Материалы».

**13.** Создайте БД агентства недвижимости. Ориентировочные таблицы-составляющие: «Объекты недвижимости», «Типы объектов», «Типы сделок», «Покупатели», «Продавцы», **«Сделки»**.

**14.** Создайте БД фирмы-распространителя программного обеспечения. Ориентировочные таблицы-составляющие: «Программное обеспечение», «Клиенты», **«Производимые работы»**, «Производители ПО», «Гарантийные сроки».

**15.** Создайте БД инструментального склада. Ориентировочные таблицы-составляющие: «Обслуживаемое оборудование», «Инструмент», **«Технологические карты»**, «Рабочие».

**16.** Создайте БД страховой фирмы. Ориентировочные таблицы-составляющие: «Виды страховок», «Клиенты», «Объекты страховки», **«Страховая деятельность»**, «Агенты».

**17.** Создайте БД подъемно-транспортного оборудования машиностроительного предприятия. Ориентировочные таблицы-составляющие: «Техника», «Производимые работы», «Виды ремонта», «Рабочие», **«Учет работ»**.

**18.** Создайте БД музея. Ориентировочные таблицы-составляющие: «Экспонаты», «Авторы», «Виды экспозиции», «Жанры», **«Проведение выставки»**.

**19.** Создайте БД НИИ. Ориентировочные таблицы-составляющие: «Сотрудники», «Научно-исследовательские разработки», **«Составление штатного расписания»**, «Отделы», «Должности».

**20.** Создайте БД студии видеозаписи. Ориентировочные таблицы-составляющие: «Режиссеры», «Актеры», «Фильмы», «Жанры», **«Продажи»**.

**21.** Создайте БД деканата. Ориентировочные таблицы-составляющие: «Список курсов и групп», «Список студентов», **«Сдача сессии»**, «Предметы», «Преподаватели».

**22.** Создайте БД автозаправочной станции. Ориентировочные таблицы-составляющие: «Горюче-смазочные материалы (ГСМ)», «Поставщики», **«Накладные»**, «Рабочие АЗС».

**23.** Создайте БД выставки продукции. Ориентировочные таблицы-составляющие: «Продукция», «Предприятие», «Выставочное место», **«Организация выставки»**.

**24.** Создайте БД полиграфической фирмы. Ориентировочные таблицы-составляющие: «Материалы», «Техника», **«Контракты»**, «Заказчики», «Перечень работ».

**25.** Создайте БД фирмы-поставщика медицинской техники. Ориентировочные таблицы-составляющие: «Ассортимент», «Заказчики», **«Контракты»**, «Производители».

**26.** Создайте БД планово-финансового отдела. Ориентировочные таблицы-составляющие: «Участки», «Работники», **«Наряды»**, «Мероприятия», «Должности».

**27.** Создайте БД строительной фирмы. Ориентировочные таблицы-составляющие: «Объекты», «Этапы выполнения», «Стройматериалы», «Бригады», **«Строительство»**.

**28.** Создайте БД опытного цеха. Ориентировочные таблицы-составляющие: «Производимые изделия», «Оборудование», **«Технологические карты»,** «Рабочие», «Инвентарь».

**29.** Создайте БД механообрабатывающего цеха. Ориентировочные таблицы-составляющие: «Детали», «Материалы», «Оснастка», **«Работа цеха»**, «Рабочие».

**30.** Создайте БД фирмы по установке и обслуживанию ЛВС. Ориентировочные таблицы-составляющие: «Сетевые конфигурации», «Клиенты», **«Договора»**, «Оборудование», «Исполнители».

**31.** Создайте БД фирмы по продаже стройматериалов. Ориентировочные таблицы-составляющие: «Сырье», «Продукция», «Заказчики», **«Контракты»**.

**32.** Создайте БД ветеринарной станции. Ориентировочные таблицы-составляющие: «Ветеринарные работы», «Обслуживаемые хозяйства», «Медикаменты», «Ветеринары», **«Проведенные работы»**.

**33.** Создайте БД фирмы по проведению буровых работ. Ориентировочные таблицы-составляющие: «Скважины», «Клиенты», «Бригады», «Оборудование», **«Контракты»**.

**34.** Создайте БД загрузки аудиторий. Ориентировочные таблицы-составляющие: «Аудитории», «Учебные дисциплины», «Преподаватели», «Кафедры», «Группы», **«Загрузка аудиторий»**.

**35.** Создайте БД трикотажной фабрики. Ориентировочные таблицы-составляющие: «Сырье», «Изделия», **«Сбыт изделий»**, «Покупатели», «Размеры».

**36.** Создайте БД ателье головных уборов. Ориентировочные таблицы-составляющие: «Изделия», «Клиенты», «Материалы», «Размеры», **«Квитанции»**.

**37.** Создайте БД гостиницы. Ориентировочные таблицы-составляющие: «Номера», «Клиенты», «Обслуживающий персонал», **«Счета»**, «Услуги».

**38.** Создайте БД жилищного коммунального хозяйства. Ориентировочные таблицы-составляющие: «Специалисты», «Жилищный фонд», «Мероприятия», **«Заказы»**.

**39.** Создайте БД стоматологической поликлиники. Ориентировочные таблицы-составляющие: «Врачи», «Пациенты», **«Обслуживание»**, «Услуги».

**40.** Создайте БД сборочного процесса. Ориентировочные таблицы-составляющие: «Комплектующие», «Изделия», **«Сборка»**, «Технологические процессы».

**41.** Создайте БД мебельной фабрики. Ориентировочные таблицы-составляющие: «Изделия», «Материалы», «Заказчики», «Поставщики», **«Контракты»**.

**42.** Создайте БД кабельного завода. Ориентировочные таблицы-составляющие: «Сырье», «Продукция», «Технологические процессы», **«Производство»**, «Рабочие».

**43.** Создайте БД механизированной колонны. Ориентировочные таблицы-составляющие: «Техника», «Выполняемые работы», **«Путевки»**, «Водители».

**44.** Создайте БД санатория. Ориентировочные таблицы-составляющие: «Оздоровительные программы», «Отдыхающие», «Врачи», **«Обслуживание»**, «Заболевания».

**45.** Создайте БД геологоразведочной экспедиции. Ориентировочные таблицы-составляющие: «Регионы», «Карты», **«Экспедиции»**, «Геологи».

**46.** Создайте БД фирмы по автоматизации производства. Ориентировочные таблицы-составляющие: «Каталог устройств и программного обеспечения», «Клиенты», «Персонал», «Услуги», **«Договора»**.

# Задание № 2

**Создание запросов**

Цель настоящего задания– изучение назначения запросов SQL, возможностей задания, а также ограничений задания запросов для безошибочной их обработки СУБД.

**Теоретические сведения**

Для извлечения информации из базы данных в SQL используется инструкция SELECT, которая возвращает искомую информацию в виде таблицы результатов запроса. Для демонстрации примеров задания инструкции предположим существование небольшой базы данных пригодной для удовлетворения информационных нужд некоторого малого предприятия, занимающегося сдачей в аренду недвижимости. Реляционная схема данной базы приведена в Приложении А.

На рисунке 7 приведена синтаксическая диаграмма инструкции select. Инструкция состоит из шести предложений. Предложения select и from являются **обязательными**. Четыре остальных включаются в инструкцию только при необходимости. Ниже перечислены функции каждого из предложений.

В предложении select указывается список столбцов, которые должны быть возвращены инструкцией select. Возвращаемые столбцы могут содержать значения, извлекаемые из столбцов таблиц базы данных, или значения, вычисляемые во время выполнения запроса.

В предложении from указывается список таблиц, которые содержат элементы данных, извлекаемые запросом.

Предложение where показывает, что в результаты запроса следует включать только некоторые строки. Для отбора строк, включаемых в результаты запроса, используется *условие отбора.* Основные возможности этого предложения описаны ниже.

Предложение group by позволяет создать итоговый запрос. Обычный запрос включает в результаты запроса по одной записи для каждой строки из таблицы. Итоговый запрос, напротив, вначале группирует строки базы данных по определенному признаку, а затем включает в результаты запроса одну итоговую строку для каждой группы.

Предложение having показывает, что в результаты запроса следует включать только некоторые из групп, созданных с помощью предложения group by. В этом предложении, как и в предложении where, для отбора включаемых групп используется условие отбора.

Предложение order by сортирует результаты запроса на основании данных, содержащихся в одном или нескольких столбцах. Если это предложение не указано, результаты запроса не будут отсортированы.



Рис 7

SQL-запросы, извлекающие из таблицы все строки, полезны при просмотре базы данных и создании отчетов, однако редко применяются для чего-нибудь еще. Обычно требуется выбрать из таблицы несколько строк и включить в результаты запроса только их. Чтобы указать, какие строки нужно отобрать, следует воспользоваться предложением WHERE.

Предложение WHERE состоит из ключевого слова WHERE, за которым следует условие отбора, определяющее, какие именно строки требуется извлечь. Если условие отбора имеет значение TRUE, строка будет включена в результаты запроса. Если же оно имеет значение FALSE или NULL, то строка исключается из результатов запроса.

В SQL обычно используются пять условий отбора:

- *сравнение*

В SQL имеется шесть различных способов сравнения выражений ( =, !=(не равно), <, >, <=, >=). Следует также отметить, что условие сравнение можно делать и составным, объединяя простые сравнения логическими предикатами AND и OR.

- *проверка на принадлежность диапазону*

Осуществляется посредством оператора BETWEEN … AND. При этом проверяется, находится ли элемент данных между двумя заданными значениями.

- *проверка на членство в множестве*

Осуществляется посредством оператор in. В этом случае проверяется, соответствует ли элемент данных какому-либо значению из заданного списка.

- *проверка на соответствие шаблону*

При применении, справа от оператора *LIKE* используют строки –шаблоны для поиска.

- *проверка на равенство значению NULL.*

Осуществляется посредством операторов *IS NULL* и *IS NOT NULL*.

Проиллюстрируем вышесказанное примерами:

*select*fname, lname

*from*staff

*where* DOB *between* ’01.10.85’ *and* ’31.12.90’;

Cледует отметить, что проверка на принадлежность избыточна в SQL, поскольку её можно выразить в виде двух сравнений, т. е. выражение A *between* B *and* C эквивалентно (A>=B) *and* (A<=C).

*select* address, tel\_no

*from* branch

*where* city *in* (‘Минск’, ‘Витебск’);

Проверка IN , также является избыточной, так как условие X IN (A, B) полностью эквивалентно условию (X=A) OR (X=B)

*select* lname, address, tel\_no

*from* staff

*where* lname *like* ‘K%’

Здесь следует обратить внимание на запись шаблона строки сравнения в условии LIKE. Указанный способ задания строки шаблона характерен для диалекта SQL СУБД ORACLE и отличен от регламентированного стандартом. Символ ‘%’ - замещает произвольную последовательность символов, а ‘\_’ – замещает одиночный символ.

ЗАПРОСЫ С ОБЪЕДИНЕНИЕМ ТАБЛИЦ

Чтобы понять, как в SQL реализуются многотабличные запросы, лучше всего начать с рассмотрения простого запроса, который объединяет данные из двух различных таблиц.

*Составить список всех сотрудников, работающих в Минском отделении.*

*select* fname, lname, position, S.tel\_no

*from* Branch B, Staff S

*where* B.bno=S.bno *and* city = ‘Минск’;

Процесс обработки приведенного двухтабличного запроса проиллюстрируем рисунком 8.



Рис. 8

Как видно из запроса в объединении по равенству используются полные имена столбцов. Здесь это – необходимость, так как при проектировании имена столбцов с идентификаторами отделов выбраны одинаковыми. Особенностью многотабличных запросов является также и то, что в предложении FROM часто используются псевдонимы таблиц, чтобы упростить полные имена столбцов, а также обеспечить однозначность ссылок на столбцы.

ЗАПРОСЫ С ГРУППИРОВКОЙ

Запрос, включающий в себя предложение GROUP BY, называется запросом с группировкой, поскольку он объединяет строки исходных таблиц в группы и для каждой группы строк генерирует одну строку в таблице результатов запроса. Столбцы, указанные в предложении GROUP BY, называются столбцами группировки, поскольку именно они определяют, по какому признаку строки делятся на группы. Приведем запрос с предложением GROUP BY :

*Определить сколько в среднем получают сотрудники в зависимости от занимаемой ими должности.*

*select* position, *avg*(salary)

*from* staff

*group by* position;

На рисунке 9 изображена схема выполнения этого запроса.



Рис. 9

На логическом уровне запрос выполняется следующим образом:

1. Сотрудники делятся на группы, по одной группе для каждой занимаемой должности. В каждой группе все сотрудники имеют одно и то же значение в столбце POSITION.

2. Для каждой группы вычисляется среднее значение столбца SALARY по всем строкам, входящим в группу, и генерируется одна итоговая строка результатов. Эта строка содержит значение столбца POSITION для группы и среднюю зарплату сотрудников для данной группы.

В качестве обобщения отметим, что между статистическими функциями SQL и предложением GROUP BY существует связь. Статистическая функция берет столбец значений и возвращает одно значение. Предложение GROUP BY указывает, что результаты запроса следует разделить на группы, применить статистическую функцию по отдельности к каждой группе и получить для каждой группы одну строку результатов.

Помимо использованной в примере статистической функции AVG(), которая вычисляет среднее значение по числовому столбцу, SQL поддерживает ещё четыре:

- SUM() – для вычисления суммы всех значений числового столбца-аргумента

- MIN() – определяет минимальное значение столбца

- MAX() – определяет максимальное значение столбца

- COUNT() – подсчитывает число всех определенных значений столбца любого типа

Последняя функция может использоваться в виде COUNT(\*), в этом случае она подсчитывает число строк таблицы.

**Варианты заданий**

**1.** Создайте запросы: «Доценты» (условная выборка); «Сводка количества работающих на каждой должности» (итоговый запрос); «Общий список сотрудников и кафедр с количеством контрактов по каждой позиции» (запрос на объединение), «Количество сотрудников на кафедрах по годам» (запрос по полю с типом дата).

**2.** Создайте запросы: «Залежавшийся товар» (условная выборка); «Рейтинг спроса по фирмам» (итоговый запрос); «Общий список продавцов и поставщиков с суммами продаж каждого» (запрос на объединение), «Количество продаж каждого вида товаров по месяцам текущего года» (запрос по полю с типом дата).

**3.** Создайте запросы: «Подписчики газеты "Вечерний Минск"» (условная выборка); «Количество подписок на каждое издание» (итоговый запрос); «Общий список подписчиков и почтальонов с количеством подписок у каждого» (запрос на объединение), «Суммы подписок по кварталам» (запрос по полю с типом дата).

**4.** Создайте запросы: «Список отделов, реализующих парфюмерию» (условная выборка); «Сводка продаж по отделам» (итоговый запрос); «Общий список отделов и продавцов с указанием сумм продаж по каждой позиции» (запрос на объединение), «Суммы продаж по неделям года» (запрос по полю с типом дата).

**5.** Создайте запросы: «Список сотрудников, не занятых в основном производстве» (условная выборка); «Сводка по заработной плате каждого отдела» (итоговый запрос); «Общий список отделов и должностей с количеством сотрудников по каждой позиции» (запрос на объединение), «Количество сотрудников штатного расписания по месяцам (выводить название месяца)» (запрос по полю с типом дата).

**6.** Создайте запросы: «Реализованные изделия за последнюю неделю» (условная выборка); «Сводка расхода сырья» (итоговый запрос); «Общий список изделий и сырья с указанием количества по каждой позиции» (запрос на объединение), «Суммы продаж товаров по неделям текущего месяца» (запрос по полю с типом дата).

**7.** Создайте запросы: «Чугунные отливки большого объема» (условная выборка); «Расход материала в плановом периоде» (итоговый запрос); «Общий список деталей и материалов с указанием количества по каждой позиции» (запрос на объединение), «Количество сделанных деталей с 9 до 12 часов» (запрос по полю с типом дата).

**8.** Создайте запросы: «Больные-пенсионеры» (условная выборка); «Количество больных в каждой палате» (итоговый запрос); «Общий список врачей с количеством больных и больных с количеством дней пребывания» (запрос на объединение), «Количество заболевших по годам» (запрос по полю с типом дата).

**9.** Создайте запросы: «Список оборудования для высококачественной обработки деталей» (условная выборка); «Загруженность оборудования» (итоговый запрос); «Общий список деталей и оборудования с количеством использований в картах» (запрос на объединение), «Количество деталей по месяцам текущего года» (запрос по полю с типом дата).

**10.** Создайте запросы: «Крупнейшие партии» (условная выборка), «Сводка по технике, закупленной каждым клиентом» (итоговый запрос); «Общий список клиентов и производителей с количеством продаж» (запрос на объединение), «Количество проданных товаров фирм-производителей по кварталам» (запрос по полю с типом дата).

**11.** Создайте запросы: «Наиболее загруженное оборудование» (условная выборка); «Количество часов наработки» (итоговый запрос); «Общий список оборудования и работ с количеством использований в картах» (запрос на объединение), «Количество используемого оборудования по месяцам» (запрос по полю с типом дата).

**12.** Создайте запросы: «Постоянные клиенты фирмы» (условная выборка); «Прибыль по каждому виду продукции/услуги» (итоговый запрос); «Общий список услуг и продукции с количеством договоров по каждой позиции» (запрос на объединение), «Количество договоров по неделям года» (запрос по полю с типом дата).

**13.** Создайте запросы: «Список объектов, предлагаемых к продаже» (условная выборка); «Сальдо по видам объектов» (итоговый запрос); «Общий список покупателей и продавцов с количеством сделок» (запрос на объединение), «Количество сделок по районам и по годам» (запрос по полю с типом дата).

**14.** Создайте запросы: «Список клиентов, программному обеспечению которых предстоит обновление» (условная выборка); «Сводка реализованного программного обеспечения» (итоговый запрос); «Общий список ПО с количеством продаж и производителей с количеством ПО у каждого» (запрос на объединение); «Количество дней от покупки ПО до его обновления» (запрос по полю с типом дата).

**15.** Создайте запросы: «Инструмент с высокой степенью используемости» (условная выборка); «Количество единиц инструмента для каждого оборудования» (итоговый запрос); «Общий список вида оборудования и инструментов с указанием количества» (запрос на объединение), «Количество задействованного оборудования по месяцам и видам оборудования» (запрос по полю с типом дата).

**16.** Создайте запросы: «Клиенты, застраховавшие свою жизнь за последний месяц» (условная выборка); «Сводка полученных/выплаченных сумм страховок по клиентам» (итоговый запрос); «Общий список клиентов и агентов с количеством договоров у каждого» (запрос на объединение), «Заключенные договора по кварталам за 2 последних года» (запрос по полю с типом дата).

**17.** Создайте запросы: «Работа техники высокой грузоподъемности» (условная выборка); «Сводка часов простоя единиц оборудования во внеплановом ремонте» (итоговый запрос); «Общий список рабочих с количеством работ и техники с количеством ремонтов» (запрос на объединение), «Количество поломок по годам» (запрос по полю с типом дата).

**18.** Создайте запросы: «Малоизвестные экспонаты» (условная выборка); «Выставлено работ по авторам» (итоговый запрос); «Общий список авторов с количеством работ и экспонатов с количеством экспозиций» (запрос на объединение), «Количество выставок по кварталам» (запрос по полю с типом дата).

**19.** Создайте запросы: «Сотрудники, задействованные в научно-исследовательских разработках» (условная выборка); «Выплаченная заработная плата (по отделам)» (итоговый запрос); «Общий список разработок и сотрудников с суммами затрат» (запрос на объединение), «Количество завершенных научных работ по месяцам (выводить название месяца)» (запрос по полю с типом дата).

**20.** Создайте запросы: «Оскар» (условная выборка); «Рейтинг продаж по актерам» (итоговый запрос); «Общий список режиссеров и фильмов с количеством фильмов» (запрос на объединение), «Суммы продаж по неделям текущего года» (запрос по полю с типом дата).

**21.** Создайте запросы: «Неуспевающие студенты» (условная выборка); «Средний балл по предметам» (итоговый запрос); «Общий список студентов и преподавателей с количеством изучаемых или преподаваемых предметов» (запрос на объединение), «Средний балл групп по годам обучения» (запрос по полю с типом дата).

**22.** Создайте запросы: «Закупки дизельного топлива» (условная выборка); «Затраты по видам продукции» (итоговый запрос); «Общий список поставщиков и ГСМ с количеством накладных» (запрос на объединение), «Количество накладных рабочего по месяцам текущего года» (запрос по полю с типом дата).

**23.** Создайте запросы: «Предприятия, арендовавшие наибольшие выставочные площади» (условная выборка); «Сводка по продукции/предприятиям» (итоговый запрос); «Общий список продукции с количеством выставляемой и предприятий с количеством выставок» (запрос на объединение), «Количество выставок и общее количество товаров по годам» (запрос по полю с типом дата).

**24.** Создайте запросы: «Материалы заказов малых объемов» (условная выборка); «Сводка расходов материалов» (итоговый запрос); «Общий список материалов и техники с количеством использумых в заказах» (запрос на объединение), «Количество заказчиков по кварталам» (запрос по полю с типом дата).

**25.** Создайте запросы: «Последние поступления» (условная выборка); «Рейтинг заказчиков по общим суммам контрактов» (итоговый запрос); «Общий список производителей с количеством изделий и заказчиков с количеством контрактов» (запрос на объединение), «Количество проданного ассортимента по месяцам текущего года» (запрос по полю с типом дата).

**26.** Создайте запросы: «Опасные работы» (условная выборка); «Начисление заработной платы по участкам» (итоговый запрос); «Общий список работников и участков с количеством нарядов по каждой позиции» (запрос на объединение), «Количество нарядов по дням недели (выводить названия)» (запрос по полю с типом дата).

**27.** Создайте запросы: «Текущие этапы работы» (условная выборка); «Сроки строительства объектов» (итоговый запрос); «Общий список объектов и стройматериалов с указанием количества этапов, где участвуют»(запрос на объединение), «Количество построенных объектов по кварталам» (запрос по полю с типом дата).

**28.** Создайте запросы: «Изделия/карты фрезерной обработки» (условная выборка); «Количество изделий по каждому оборудованию» (итоговый запрос); «Общий список изделий и оборудования с количеством карт, в которых используются» (запрос на объединение), «Количество занятого единиц оборудования по часам за текущий день» (запрос по полю с типом дата).

**29.** Создайте запросы: «Оснастка для обработки деталей высокой твердости» (условная выборка); «Расход материалов по видам деталей» (итоговый запрос); «Общий список материалов и оснастки с общим количеством по каждому виду» (запрос на объединение), «Количество рабочих по дням недели» (запрос по полю с типом дата).

**30.** Создайте запросы: «Список клиентов, с которыми необходимо перезаключать договора» (условная выборка); «Протяженность кабеля у каждого клиента» (итоговый запрос); «Общий список клиентов и исполнителей с количеством договоров по каждому» (запрос на объединение), «Количество договоров по сетевым конфигурациям в год» (запрос по полю с типом дата).

**31.** Создайте запросы: «Продажа кирпича» (условная выборка); «Сводка сумм контрактов по каждому заказчику» (итоговый запрос); «Общий список продукции и сырья с указанием количесва в наличии» (запрос на объединение), «Количество контрактов в год по заказчикам» (запрос по полю с типом дата).

**32.** Создайте запросы: «Список хозяйств, запланированных на ближайшую неделю» (условная выборка); «Объемы проведенных работ в каждом хозяйстве» (итоговый запрос); «Общий список обслуживаемых хозяйств и ветеринаров с количеством обслуживаемых животных» (запрос на объединение), «Количество животных по дням текущей недели» (запрос по полю с типом дата).

**33.** Создайте запросы: «Контракты на бурение глубинных скважин» (условная выборка); «Количество контрактов на каждый вид работ» (итоговый запрос); «Общий список клиентов и бригадиров с количеством контрактов у каждого» (запрос на объединение); «Общий метраж скважин за последние 10 недель» (запрос по полю с типом дата).

**34.** Создайте запросы: «Поточные (лекционные) аудитории» (условная выборка); «Рейтинг дисциплин по загрузке аудиторий» (итоговый запрос); « «Общий список кафедр и групп с указанием количества студентов» (запрос на объединение), «Количество занятий групп по дням недели» (запрос по полю с типом дата).

**35.** Создайте запросы: «Сбыт изделий, содержащих искусственные волокна» (условная выборка); «Рейтинг себестоимости изделий» (итоговый запрос); «Общий список используемого сырья и изделий с указанием количества в наличии» (запрос на объединение), «Квитанции за последние 5 дней» (запрос по полю с типом дата).

**36.** Создайте запросы: «Заказчики зимнего ассортимента» (запрос – условная выборка); «Популярность моделей» (итоговый запрос); «Общий список изделий и материалов с количеством в наличии» (запрос на объединение), «Количество изделий по годам» (запрос по полю с типом дата).

**37.** Создайте запросы: «Богатые клиенты» (условная выборка); «Рейтинг загрузки номеров» (итоговый запрос); «Общий список клиентов и персонала с количеством номеров проживания или обслуживания» (запрос на объединение), «Общий доход по месяцам за текущий год» (запрос по полю с типом дата).

**38.** Создайте запросы: «Деятельность, запланированная на ближайшую неделю» (условная выборка); «Рейтинг работников по количеству мероприятий» (итоговый запрос); «Общий список специалистов и мероприятий с количеством мероприятий по каждой позиции» (запрос на объединение), «Количество проведенных мероприятий и задействованных специалистов по месяцам» (запрос по полю с типом дата).

**39.** Создайте запросы: «Протезирование» (условная выборка); «Рейтинг услуг» (итоговый запрос); «Общий список врачей и пациентов с количеством оказанных услуг» (запрос на объединение), «Количество пациентов по дням текущего месяца» (запрос по полю с типом дата).

**40.** Создайте запросы: «Изделия, требующие микросборки» (условная выборка); «Количество комплектующих в каждом изделии» (итоговый запрос); «Общий список комплектующих и изделий с количеством в наличии по каждой позиции» (запрос на объединение), «Количество изделий по кварталам» (запрос по полю с типом дата).

**41.** Создайте запросы: «Контракты по корпусной мебели» (условная выборка); «Рейтинг продукции» (итоговый запрос); «Общий список заказчиков и поставщиков с количеством контрактов у каждого» (запрос на объединение), «Количество изделий в месяц (выводить название месяца)» (запрос по полю с типом дата).

**42.** Создайте запросы: «Монтажные провода» (условная выборка); «Затраты сырья по видам продукции» (итоговый запрос); «Общий список сырья и продукции с количеством по каждой позиции» (запрос на объединение), «Общий расход материалов по годам» (запрос по полю с типом дата).

**43.** Создайте запросы: «Экскаваторщики» (условная выборка); «Отработано персоналом часов за неделю» (итоговый запрос); «Общий список водителей и техники с количеством нарядов по каждой позиции» (запрос на объединение), «Количество водителей техники по неделям года» (запрос по полю с типом дата).

**44.** Создайте запросы: «Программы по желудочно-кишечным заболеваниям» (условная выборка); «Сводка отдыхающих за плановый период» (итоговый запрос); «Общий список врачей и отдыхающих с количеством программ у каждого» (запрос на объединение), «Количество заболеваний по месяцам» (запрос по полю с типом дата).

**45.** Создайте запросы: «Экспедиции по Беларуси» (условная выборка); «Протяженность маршрутов по регионам» (итоговый запрос); «Общий список регионов и геологов с количеством экспедиций по каждой позиции» (запрос на объединение), «Количество карт по кварталам текущего года» (запрос по полю с типом дата).

**46.** Создайте запросы: «Услуги, оказанные в текущем месяце» (условная выборка); «Сводка доходов по виду услуг за отчетный период» (итоговый запрос); «Общий список клиентов и персонала с количеством договоров каждого» (запрос на объединение), «Количество договоров по годам» (запрос по полю с типом дата).

# Задание № 3

**Создание процедур и функций на PL/SQL**

Цельнастоящего задания **–** изучение возможностей и основных программных конструкций языкаPL/SQL для разработки хранимых процедур и функций баз данных ORACLE.

**Теоретические сведения**

В своей исходной форме SQL не был полноценным языком программирования. Он задумывался и создавался как язык, предназначенный для выполнения операций над базами данных – создания их структуры, ввода и обновления данных – и особенно для выполнения запросов. SQL может использоваться как интерактивный командный язык: пользователь по очереди вводит инструкции SQL с клавиатуры, а СУБД их выполняет. В этом случае последовательность операций над базой данных определяется ее пользователем.

С появлением хранимых процедур и триггеров язык SQL обогатился рядом дополнительных базовых возможностей, обеспечиваемых практически всеми языками программирования, что позволило писать на “расширенном SQL” настоящие программы и процедуры.

*PL/SQL* — процедурный язык программирования, встроенный в большинство продуктов ORACLE. С помощью PL/SQL можно создавать программы, в которых объединяются операторы PL/SQL, предназначенные для управления выполнением программ, и SQL-операторы, с помощью которых можно обращаться к базам данных и обрабатывать информацию.

Программа PL/SQL структурируется при помощи отдельных блоков, в которых группируются связанные объявления и операторы. Каждый блок программы PL/SQL имеет специальное назначение и решает определенную задачу.

Блок PL/SQL может включать *три раздела*: объявления, тело основной программы и обработчики исключительных ситуаций:

• *Раздел объявлений* блока PL/SQL – это часть блока, где объявляются переменные, константы, исключительные ситуации и т.д., которые затем становятся доступны во всех других частях блока.

• В *теле основной программы* содержатся исполняемые операторы блока. Другими словами, тело – это часть блока PL/SQL, в которой определяются его функциональные возможности.

• В *разделе исключительных ситуаций* находятся обработчики исключительных ситуаций, т.е. подпрограммы обработки ошибок. Когда оператор в теле блока определяет исключительную ситуацию (обнаруживает ошибку), он передает управление программой обработчику соответствующей исключительной ситуации, находящемуся в разделе обработчиков исключительных ситуаций.

Таким образом, структуру блока PL/SQL можно представить в следующем виде:

*declare*

/\*раздел объявлений (переменные, типы, курсоры и др.)\*/

*begin*

/\*Основной раздел блока, называемый выполняемым. Содержит процедурные и SQL – операторы\*/

*exception*

/\*Раздел обработки исключительных ситуаций. Содержит операторы обработки ошибок\*/

end;

Вообще говоря, **обязательным** в блоке является только выполняемый раздел, кроме того, он должен содержать, по крайней мере, один выполняемый оператор.

Программы PL/SQL не ограничены одним блоком. Для лучшей организации выполнения подзадач в программе PL/SQL можно создать вложенные блоки в теле программы и в разделе исключительных ситуаций блока. Вообще говоря, объявления блока видны всем операторам тела блока и раздела обработки исключительных ситуаций, в том числе и операторам “вложенных подблоков”.

В разделе объявлений программы PL/SQL могут содержаться объявления переменных и именованных констант. Переменные и константы могут иметь любой тип данных ORACLE.

Важной особенностью программ PL/SQL является возможность использования атрибутов %TYPE и %ROWTYPE для того, чтобы объявлять переменные, соответствующие столбцам (%TYPE) и таблицам баз данных (%ROWTYPE). Использование атрибутов не только упрощает объявление программных конструкций, но и делает программы более удобными для модификации баз данных. Например, после того как администратор модифицирует таблицу и добавляет новый столбец, переменная типа записи, объявленная с атрибутом %ROWTYPE, в процессе работы автоматически регулируется и учитывает новый столбец.

*declare*

pos staff.position *%type*; /\* объявление переменной pos по “образцу” столбца position таблицы staff \*/

st staff *%rowtype*; /\* объявление переменной–записи st по “образцу” таблицы staff \*/

*begin*

st.salary:=500;/\* использование поля salary записи st \*/

select sysdate into st.dob from dual;/\* использование поля dob записи st\*/

…

*end*;

Для взаимодействия с базами данных в основном разделе блока PL/SQL можно воспользоваться оператором SELECT INTO , как в вышеприведенном примере. Однако применение этого оператора ограничено, так как если в результирующем множестве оператора SELECT INTO содержится более одной строки, то ORACLE возвращает сообщение об ошибке. Для обработки запроса, возвращающего более одной строки, в программах используют курсоры.

Перед тем как использовать курсор, он должен быть объявлен в разделе объявлений блока программы, а также переменные для считывания данных из результирующего множества курсора после его открытия.

*declare*

*cursor* cur *is* *select* position, *avg*(salary) *as* sal *from* staff *group by* position;

cur\_val cur *%rowtype*; /\*объявление переменной для считывания\*/

…

*end*;

Далее для работы со строками, извлекаемыми из результирующего множества курсора необходимо выполнить три операции: открыть курсор (OPEN), считать информацию в ранее объявленные переменные (FETCH) и закрыть курсор (CLOSE). Считывание происходит в цикле до последней (включительно) строки результирующего множества курсора. Для выхода из цикла считывания можно использовать курсорные атрибуты %FOUND и %NOTFOUND .

Работу с курсорами существенно упрощают курсорные циклы FOR. В курсорном цикле FOR автоматически объявляется переменная для считывания, открывать курсор, выбирать из него строки и закрывать курсор, когда из него выбирается последняя строка.

*begin*

*for* val *in* (*select* position, *avg*(salary) *as* sal *from* staff *group by* position)

*loop*

DBMS\_OUTPUT.PUT\_LINE(val.position ||’ ‘|| val.sal);

*end loop*;

…

*end;*

Программу нельзя считать завершенной, если она не содержит подпрограмм, предназначенных для обработки ошибок, происходящих во время выполнения обычных операций по обработке данных. В программах PL/SQL подпрограммы обработки ошибок в тело программы не встроены. Они представляют собой так называемые обработчики исключительных ситуаций. Распознав ошибку, программа PL/SQL устанавливает (*raises*) исключительную ситуацию (*exception*) и передает управление соответствующей подпрограмме – обработчику исключительных ситуаций, которая не является частью тела программы.

Исключительная ситуация представляет собой поименованное условие возникновения ошибки. В PL/SQL имеется множество заранее определенных исключительных ситуаций, соответствующих чаще всего встречающимся в ORACLE ошибкам. Например:

• Программа распознает исключительную ситуацию NO\_DATA\_FOUND (данные не обнаружены) в случае, если в результирующем множестве оператора SELECT INTO нет строк, и исключительную ситуацию TOO\_MANY\_ROWS (слишком много строк), когда в результирующем множестве оператора SELECT INTO более одной строки.

• Программа распознает исключительную ситуацию DUP\_VAL\_ON\_INDEX (повторяющееся значение в индексе) в случае, если оператор INSERT или UPDATE дублирует ключевое значение, уже находящееся в таблице.

• Программа распознает исключительную ситуацию ZERO\_DIVIDE (деление на нуль) в случае, если оператор пытается поделить число на нуль.

В PL/SQL включено почти 20 заранее определенных исключительных ситуаций. Когда в программе встречается заранее определенная исключительная ситуация, то управление программой автоматически передается соответствующему обработчику исключительных ситуаций, если таковой имеется.

В PL/SQL возможно использование “собственных” исключительных ситуаций, т.е. не предустановленных, а определяемых разработчиком. Однако чтобы затем установить такую исключительную ситуацию, необходимо выполнить ее явную проверку.

*declare*

invalid\_person *exception*;

err\_num *integer*;

err\_msg *varchar*(2000);

*begin*

*select* … *into* … *from*…;

*update* staff

*set* salary=salary\*1.1;

*where* sno=101;

*if* SQL %NOTFOUND *then* /\* проверка возникновения пользовательской исключительной ситуации. %NOTFOUND – курсорный атрибут \*/

*raise* invalid\_person; /\* установка пользовательской исключительной ситуации \*/

*end if*;

*exception*

*when no\_data\_found then*

*raise\_application\_error* (-20001, ‘Строки не выбраны’);

*when too\_many\_rows then*

*raise\_application\_error* (-20002, ‘Слишком много строк ’);

*when* invalid\_person *then*

*raise\_application\_error* (-20003, ’Неверный идентификатор сотрудника’);

*when others then*

err\_num:=SQLCODE;

err\_msg:=SUBSTR(SQLERRM, 1, 100);

*raise\_application\_error*(-20000, err\_num || ‘ ‘||err\_msg);

*end*;

В данном примере:

- Для возвращения в вызывающую среду номера и сообщения об ошибке, определяемой пользователем, применяется процедура *raise\_application\_error* (установить ошибку приложения). Номера сообщений об ошибках, определяемых пользователем должны лежать в диапазоне от -20000 до -20999.Что касается обработчиков “встроенных” исключительных ситуаций, то это по сути “перехватчики” стандартных реакций на возникшие ошибки. Как видно из примера, они заменяют стандартные сообщения об ошибках на более “понятные”.

- Для создания общего обработчика всех исключительных ситуаций, для которых не определены собственные обработчики, применяется синтаксис *when others*.

- Для возвращение номера и сообщения о самой последней ошибке ORACLE применяются специальные функции SQLCODE и SQLERRM.

ТИПЫ ПРОГРАММ PL/SQL

Существует три типа программ PL/SQL: анонимные блоки, хранимые процедуры, хранимые функции.

Анонимный блок – это блок PL/SQL, не хранящийся в базе данных. В процессе работы приложение просто посылает такой кодовый блок серверу баз данных, и после его обработки сервером блок прекращает существование.

Хранимые процедуры и функции. – это именованные программы PL/SQL, которые могут использовать параметры и вызываться приложением. Скомпилированный двоичный код программ можно хранить в базе данных. Хранимые процедуры и функции в отличие от анонимных блоков сохраняются в базе данных и наряду с таблицами являются самостоятельными объектами баз данных. Процедуры и функции сохраняются в базе данных в откомпилированном виде, и при их вызове загружаются в разделяемый пул (*pool* – системная область памяти СУБД), откуда удаляются по мере его заполнения в порядке частоты использования кода процедуры или функции. Наиболее редко используемый код, разумеется, удаляется раньше, и при очередном вызове данный код снова загружается в пул с диска. Хранимая функция отличается от процедуры тем, что функция в отличие от процедуры возвращает значения в вызывающую среду.

Для создания в базе данных ORACLE хранимой процедуры или функции применяется команда CREATE PROCEDURE или CREATE FUNCTION:

*сreate*[*or replace*] *procedure* имя\_процедуры [(аргумент1 [{*in* | *out* | *in out*}] тип, … аргумент2 [{*in* | *out* | *in out*}] тип)] {*is*|*as*} тело процедуры/

*сreate*[*or replace*] *function* имя\_функции [(аргумент1 [{*in* | *out* | *in out*}] тип, … аргумент2 [{*in* | *out* | *in out*}] тип)] *return* возвращаемый тип{*is*|*as*} тело функции оператор *return*/

Чтобы изменить текст процедуры или функции, её необходимо удалить, а затем повторно создать. Во время разработки процедур или функций эта операция повторяется достаточно часто, поэтому ключевые слова OR REPLACE позволяют выполнить такую операцию за один раз. Если процедура или функция существует, то она удаляется без всякого предупреждения (в данном случае вызов инструкции DROP PROCEDURE / DROP FUNCTION не требуется), если же не существовала, то она просто создаётся.

Параметры IN, OUT и IN OUT используются как при создании процедур, так и функций. Смысл параметра INв следующем: Значение физического параметра передаётся в подпрограмму. Внутри подпрограммы формальный параметр рассматривается в качестве константы PL/SQL (параметр только для чтения) и не может быть изменен. Когда подпрограмма завершается, и управление передаётся в вызывающую среду, фактический параметр не изменяется. При использовании параметра OUT, любое значение, которое имеет фактический параметр при вызове, игнорируется. Внутри вызванной подпрограммы формальный параметр рассматривается в качестве неинициализированной переменной, т. е содержит NULL – значение, и можно как записать в него значение, так и считать значение из него. Когда управление передаётся в вызывающую среду, содержание формальной переменной присваивается фактическому параметру. Параметр IN OUT это комбинация параметров INи OUT*.* В данном случае формальный параметр рассматривается в качестве инициализированной переменной

ВЫЗОВ ХРАНИМЫХ ПРОЦЕДУР И ФУНКЦИЙ

Хранимую процедуру можно вызывать в основном разделе блока PL/SQL, в том числе в блоках других процедур или функций. Вызов функций может осуществляться либо в блоке посредством оператора присваивания, либо прямо в SQL – операторе.

*create or replace procedure* sal\_up (fst *in varchar*, lst *in varchar*) *is*

*begin*

*update* staff *set* salary=salary\*1.1 *where* fname=fst *and* lname=lst;

*end*;

/\*вызов хранимой процедуры sal\_up в анонимном блоке \*/

*begin*

sal\_up(‘Иван’,’Иванов’);

*end*;

/\* Создание хранимой функции\*/

*create or replace function* st\_id (fst *in varchar*, lst *in varchar*) *return number is*

id *number*;

*begin*

*select* sno *into* id *where* fname=fst *and* lname=lst;

*return* id;

*exception*

*when* NO\_DATA\_FOUND *then*

*raise\_application\_error*(-20001,’Сотрудник не найден’);

*when* TOO\_MANY\_ROWS *then*

*raise\_application\_error*(-20002,’У нас работает не один сотрудник’ || fst || lst);

/\*Вызов хранимой функции st\_id из SQL\*/

*update* staff *set* salary=salary\*1.1 *where* sno=st\_id(‘Иван’,’Иванов’);

select st\_id(‘Иван’,’Иванов’) *from dual*;

**Варианты заданий**

**1.** Написать процедуру, которая повышает заработную плату сотрудников:

* до размера в 1,5 прожиточного минимума, если оклад меньше прожиточного минимума;
* на 5%, если оклад больше, чем 1,5 прожиточного минимума;

Входной параметр – прожиточный минимум. Выводить количество обновленных записей.

Создать функцию, которая определяет, работает ли тот или иной сотрудник на указанной кафедре (путем возвращения или FALSE с соответствующим комментарием), а в параметре *out* при выходе из функции в случае TRUE записывает номер телефона сотрудника.

**2.** Написать процедуру, которая формирует заказ на товары, которых осталось меньше необходимого запаса и выводит его на экран, подсчитывая общий итог. Величину минимального запаса передавать во входном параметре.

Создать функцию, возвращающую пустую строку или строку «закончился срок реализации», если дата поставки плюс срок реализации партии товара больше текущей даты и времени. Поля даты поставки и срока реализации должны присутствовать в таблицах БД.

**3.** Создать процедуру, копирующую строки с информацией о подписках за текущий месяц, во вспомогательную таблицу. Все коды в таблице подписок должны быть заменены на реальные данные из связанных таблиц.

Написать функцию, которая выводит информацию о подписчиках, которые оформили количество подписок за указанный год большее, чем указано во входном параметре. Возвращать общее количество подписчиков.

**4.** Создать процедуру, распечатывающую список товаров с их общим количеством продаж, старой и новой ценой (для товаров, которые не продавались в течение месяца снизить цену на 10%) по отделам:

|  |  |  |  |
| --- | --- | --- | --- |
| Отдел 1 |  |  |  |
| товар1 | количество | старая\_цена | новая\_цена |
| товар2 | количество | старая\_цена | новая\_цена |
| Отдел 2 |  |  |  |
| товар1 | количество | старая\_цена | новая\_цена |
| товар2 | количество | старая\_цена | новая\_цена |

Написать функцию, которая возвращает общее количество продавцов, работающих в отделах. Вывести также количество продавцов по отделам и продавцов, которые еще не участвовали в продажах.

**5.** Создать процедуру расчета общего количества отработанных часов в виде:

|  |  |
| --- | --- |
| Отдел 1 |  |
| фамилия1 | количество часов |
| фамилия2 | количество часов |
| Отдел 2 |  |
| фамилия1 | количество часов |
| фамилия2 | количество часов |

Написать функцию, определяющую, являются ли сотрудники указанного отдела пенсионерами по полу и дате рождения. В результате работы выводить ФИО сотрудников отдела с пометками «пенсионер» или «не достигнут пенсионный возраст», а также возвращать общее количество пенсионеров.

**6.** Создать процедуру, переносящую в архив информацию о товарах, которые реализованы ранее текущего месяца. Архив должен содержать не значения ключей, а реальные названия изделий и сырья.

Написать функцию, которая возвращает пустую строку или строку «остатки», если заданного товара осталось меньше 10 единиц, а также выводит перечень товаров с оставшимся количеством.

**7.** Создать хранимую процедуру, выводящую количество сделанных деталей заданного наименования и в случае, если заказов на эту деталь нет в течение года, то удаляет ее из исходной таблицы.

Написать функцию, которая возвращает общее количество деталей и выводит список использованных материалов с указанием количества.

**8.** Создать процедуру, выводящую список палат с указанием количества коек и статуса палаты:

«пустая», если в палате никто не лежит,

«свободных мест нет», если палата заполнена,

«мужская», если в палате лежат только мужчины,

«женская», если в палате лежат только женщины,

«смешанная» во всех остальных случаях.

Написать функцию, возвращающую строку «больше месяца», если со времени поступления пациента прошло более 1 месяца. В качестве параметра использовать диагноз, и заменять для такого пациента его лечащего врача на того, у кого меньше всего больных.

**9.** Создать процедуру, которая копирует строки с информацией об оборудовании указанного вида во вспомогательную таблицу и подсчитывает количество деталей на нем произведенного.

Создайте функцию, возвращающую количество рабочих в бригаде. Входной параметр функции – id бригады. Если в бригаде менее 3 человек, то обновить количество до 5 человек.

**10.** Написать процедуру, которая изменяет домашний адрес клиента по указанной в качестве параметра фамилии. В случае обнаружения однофамильцев выводить их количество и уточняющие данные.

Создать функцию, подсчитывающую общее количество реализованной продукции заданного производителя. Если какой-либо продукции реализовано менее 1 единицы в год, то удалить информацию об этой продукции во вспомогательную таблицу.

**11.** Создать процедуру, обеспечивающую вывод информации обо всем оборудовании, участвующем в указанной ключевым словом работе.

Написать функцию, которая возвращает строку «не руководит работами», если нет ни одного акта в настоящий момент у переданного в качестве параметра сотрудника.

**12.** Создать процедуру изменения стоимости услуги. Входные параметры – id услуги и новая стоимость. Вывести отчет о том какая именно услуга подорожала или подешевела и на сколько %.

Написать функцию, которая за указанный период определяет количество заключенных договоров. В качестве параметра передать начальную и конечную даты периода.

**13.** Написать процедуру изменения мобильного номера продавца по указанной в качестве параметра фамилии. Контролировать, чтобы повторно не был введен тот же номер.

Создать функцию, подсчитывающую количество сделок, сделанных потенциальными покупателями за текущий день. В вызывающую среду возвращать объекты недвижимости, участвующие в этих сделках.

**14.** Создать процедуру изменения стоимости ПО. Входные параметры – производитель программного продукта и новая стоимость. Вывести количество обновленных записей.

Написать функцию, подсчитывающую количество клиентов, которым предстоит обновить указанное ПО в ближайшие два месяца.

**15.** Написать процедуру, создающую отчет о движении инструментов за последний период:

Отчет за период с … по …

1. инструмент1 поступление (ед.изм.) количество используемого

2. инструмент 2 поступление (ед.изм.) количество используемого

…

Параметр: длительность периода (в днях).

Создать функцию, возвращающую пустую строку или строку «остатки», если инструмента на складе осталось меньше 100 единиц, а также выводит инструмент с минимальным количеством в остатке.

**16.** Создать процедуру, увеличивающую на заданный % зарплату агентам, которые заключили наибольшее количество страховых договоров.

Написать функцию, которая возвращает количество клиентов, у которых срок страхования подходит к концу. Вывести список клиентов и дату окончания контракта.

**17.** Написать процедуру, которая за указанный период определяет для указанного вида техники количество и выводит перечень выполненных работ. В качестве параметров передать идентификатор техники, начальную и конечную даты периода. Результаты занести в специальную таблицу.

Создать функцию, которая возвращает количество оборудования, находящегося в ремонте и выводит информацию о сроках его окончания.

**18.** Написать процедуру, выводящую список экспонатов автора, у которого имеется не менее 5 работ. Если таких авторов несколько, то вывеси их фамилии и количество экспонатов.

Создать функцию, которая возвращает количество запланированных выставок в заданном месяце.

**19.** Создать процедуру, «переводящую» сотрудников заданного отдела в другие отделы этого же города. В каждый из отделов переводить по приблизительно одинаковому, в смысле среднего значения, количеству сотрудников. Расформированный отдел удалить.

Написать функцию, подсчитывающую количество сотрудников, работающих в заданном отделе.

**20.** Написать процедуру, выводящую список фильмов, в которых режиссер является одновременно исполнителем одной из главных ролей, с указанием фамилии режиссера и роли, которую он сыграл.

Создать функцию, которая возвращает продолжительность фильма в виде строки "Х ч. Y м." по значению числового поля «Продолжительность».

**21.** Написать функцию, которая возвращает время начала и завершения экзамена или консультации. Использовать 2 входных параметра: дата сдачи и тип (0 – экзамен, 1 – консультация). Продолжительность экзамена – 5 часов, консультации – 2 часа.

Создать процедуру, которая выводит расписание сессии. Список должен выглядеть приблизительно так:

Факультет\_1

группа1

дисципл1 препод. конс. (дата, время, ауд.) экз. (дата, время, ауд.)

дисципл2 препод. конс. (дата, время, ауд.) экз. (дата, время, ауд.)

дисципл3 препод. конс. (дата, время, ауд.) экз. (дата, время, ауд.)

группа2

дисципл1 препод. конс. (дата, время, ауд.) экз. (дата, время, ауд.)

дисципл2 препод. конс. (дата, время, ауд.) экз. (дата, время, ауд.)

…

Факультет\_2

…

**22.** Создать процедуру, обеспечивающую удаление поставщика из таблицы по указанному названию и городу. Накладные удаленного поставщика перенести во вспомогательную таблицу.

Написать функцию, которая возвращает наиболее востребованный за последний месяц вид ГСМ.

**23.** Написать процедуру, которая принимает в качестве параметра номер выставочного места и выводит список всех предприятий или товаров, принимающих участие в выставке.

Создать функцию, возвращающую количество дней, прошедших между датой последней выставки и сегодняшней датой. Если первая дата больше второй, функция возвращает -1.

**24.** Написать процедуру, которая выводит информацию о контрактах, в которых сумма заказа лежит в диапазоне +/- 50$ от введенного значения. Если контрактов с такой суммой не имеется, должно выводиться соответствующее сообщение.

Создать функцию, подсчитывающую количество заключенных контрактов за текущий год. В вызывающую среду возвращать общую сумму этих контрактов в параметре out.

**25.** Создать процедуру изменения стоимости медицинской техники. Входные параметры – id медицинской техники из таблицы «Ассортимент» и новая стоимость. Подсчитать и вывести на сколько % изменилась стоимость.

Написать функцию, подсчитывающую количество и сумму контрактов медицинской техники заданного производителя. Если указанного производителя нет, то добавить его.

**26.** Создать процедуру, копирующую строки с информацией об участках, нарядов на которые в указанном месяце не было во вспомогательную таблицу. Вывести количество таких участков.

Создать функцию, возвращающую количество работников, которые получили наименьшее количество нарядов до указанной в параметре даты. Вывести более подробную информацию об этих работниках.

**27.** Создать процедуру, копирующую строки с информацией о строительстве в текущем месяце во вспомогательную таблицу. Подсчитать количество извлеченных строк.

Создать функцию, подсчитывающую, сколько этапов выполнено по каждому объекту. Вернуть количество объектов, по которым выполнены все этапы.

**28.** Написать процедуру замены на текущий месяц заболевшего рабочего в технологических картах на другого, который меньше других загружен.

Создать функцию, возвращающую количество произведенных изделий в текущем месяце. Входной параметр функции – идентификатор изделия.

**29.** Написать процедуру определения занятости указанного рабочего за текущий год в % от общего количества выполненной работы.

Создать функцию, определяющую количество материалов в среднем расходующегося в месяц в цехе.

**30.** Создать процедуру, обеспечивающую удаление договоров, заключенных на фамилию клиента, указанную в качестве параметра.

Написать функцию, которая за указанный период определяет количество заключенных договоров. В качестве параметров передавать начальную и конечную даты периода, контролировать правильность введенных значений.

**31.** Создать процедуру, которая заменяет сырье в еще не выполненных контрактах и пересчитывает их общую стоимость. В качестве параметров предавать новое сырье и сырье, которое надо заменить.

Написать функцию, которая за указанный месяц определяет сумму подписанных контрактов. В качестве параметра передать название месяца в текстовом виде.

**32.** Создать процедуру, которая при передаче в качестве параметра, например, 1, заполняет таблицу с проведенными работами профилактическим осмотром всех хозяйств (в день не более 2 хозяйств), а при получении в качестве параметра, например, 2, подсчитывает количество работ за текущий месяц.

Написать функцию, которая определяет, работает тот или иной ветеринар на ветеринарной станции (путем возвращения TRUE или FALSE), а в параметре out по выходе из функции записывает его номер телефона.

**33.** Создать процедуру, обеспечивающую удаление клиента из таблицы по указанным имени и фамилии, а все его договора перемещает в архив.

Написать функцию, которая определяет наиболее загруженную бригаду и, если есть незанятая на текущий момент времени бригада, то распределяет задачи между ними. Функция возвращает ФИО бригадира загруженной бригады, а также выводит сообщение о том, какие проведены перестановки.

**34.** Создать процедуру, обеспечивающую вывод информации о всех аудиториях, которые свободны в указанный в качестве параметра день.

Написать функцию, которая подсчитывает количество часов занятий в неделю для указанной группы.

**35.** Создать процедуру изменения стоимости изделий. Входные параметры – идентификатор изделия и новая стоимость.

Создать функцию, возвращающую идентификатор изделия, имеющего наибольший спрос.

**36.** Создать процедуру, обеспечивающую удаление клиентов из таблицы, если они не обращались в ателье более года.

Создать функцию, возвращающую название изделия, имеющего наименьший спрос.

**37.** Написать процедуру переноса данных об услугах, оказанных выбывшим клиентам, в архив (вспомогательную таблицу). Если в один день клиенту было оказано несколько одинаковых услуг, в архивную таблицу добавлять их одной строкой, указывая количество.

Создать функцию, возвращающую количество дней, прожитых постояльцем в гостинице на основании двух дат (день приезда – день отъезда считать, как один день). Если второй параметр не определен, считать до текущей даты.

**38.** Создать процедуру, переносящую информацию о заказах, поступивших в указанный день, во вспомогательную таблицу, подсчитывать их общее количество

Написать функцию, которая подсчитывает среднее количество работ по переданной в качестве параметра должности специалиста.

**39.** Создать процедуру изменения стоимости услуги. Входные параметры – идентификатор услуги и новая стоимость. Определить на сколько в % изменилась стоимость услуги.

Создать функцию, которая возвращает ФИО врачей, у которых нет пациентов на указанную в параметре дату, если указан выходной день, то сообщать об этом.

**40.** Создать процедуру, подсчитывающее количество комплектующих в заданном изделии. Выводить изделие и количество комплектующих в %.

Написать функцию, возвращающую количество изделий, сборка которых на данный момент не завершена.

**41.** Написать процедуру изменения адреса поставщика по указанному названию. Адрес не должен повторяться, старый адрес записывать во вспомогательную таблицу с указанием текущей даты.

Написать функцию, которая за указанный период определяет количество подписанных контрактов. В качестве параметра передать начальную и конечную даты периода.

**42**. Создать процедуру, которая проверяет, выполняется ли план производства за месяц. Значение плана передавать в процедуру в виде общего количества деталей.

Написать функцию, которая вычисляет общее время технологических процессов для указанного типа кабеля.

**43.** Создать процедуру, обеспечивающую вывод информации обо всей технике, которая не работает на текущую дату и подсчитывает время простоя.

Написать функцию, которая определяет, работал ли указанный водитель на указанной технике, а в параметре out по выходе из функции записывает количество раз.

**44.** Создать процедуру, переносящую информацию об отдыхающих, с указанным в качестве аргумента заболеванием, во вспомогательную таблицу.

Создать функцию, возвращающую количество дней, прожитых отдыхающим в санатории на основании двух дат (день приезда – день отъезда считать, как один день). Если второй параметр не определен, считать до текущей даты.

**45.** Создать процедуру, переносящую информацию о геологах, участвовавших в определенной экспедиции, во вспомогательную таблицу. Аргументом является название экспедиции.

Написать функцию, которая определяет, сколько экспедиций было по указанной карте (региону).

**46**. Создать процедуру, которая подсчитывает количество договоров у персонала и на основании этой информации вычисляет их заработную плату. Результаты вычисления занести в новую таблицу.

Написать функцию, которая возвращает % скидки клиента в зависимости от суммы предыдущих договоров. (предусмотреть не менее 3 позиций).

# Задание № 4

**Создание триггеров**

Цель настоящего задания **–** изучение предназначения и особенностей создания триггеров баз данных. Определение особенностей данных объектов баз данных от хранимых процедур и функций.

**Теоретические сведения**

Триггер базы данных – это хранимая процедура, которую можно связать с некоторой таблицей. Когда приложение выполняет SQL оператор DML над таблицей, удовлетворяющей условиям, указанным в триггере, ORACLE автоматически активизирует или выполняет триггер. Таким образом, триггеры можно использовать для настройки реакции ORACLE на различные события приложения. Общий синтаксис создания триггеров, таков:

*сreate* [*or replace*] *trigger* имя\_триггера

{*before* | *after* | *instead of*} активизирующее\_событие

*on* имя\_таблицы

[*for each row*]

тело\_триггера

Здесь активизирующее\_событие, - это событие активации триггера (далее указывается конкретная таблица или представление).

Например, следующий триггер автоматически регистрирует изменения, вносимые в таблицу staff:

*create or replace* *trigger* staff\_log

*after insert or update or delete on* parts

*declare*

stmt\_type *char*(1);

*begin*

*if inserting* *then*

stmt\_type:=’I’;

*elsif updating then*

stmt\_type:=’U’;

*else*

stmt\_type:=’D’;

*end if*;

*insert into* part\_change\_log

*values*(stmt\_type, USER);

*end* ;

Как видно из выше приведенного примера, описание триггера содержит однозначно определяемые части:

- Список операторов, активизирующих триггер, который включает INSERT, UPDATE и/или DELETE. Триггер может быть связан с одной и только с одной таблицей

- Триггер может быть активизирован до (*before*) или после (*after*) выполнения оператора триггера в зависимости от логики конкретного приложения.

Помимо этого в описании триггера, указывается должен ли триггер активизироваться один раз, независимо от того, на какое количество строк оказывает воздействие оператор триггера (такие триггеры носят название *операторных*), или для каждой изменяемой строки (такие триггеры называются *строчными*). Приведенный выше триггер это операторный триггер. В следующем примере триггер является строчным:

*create or replace trigger* staff\_upd

*before insert or update*

*on* staff

*for each row*

*begin*

/\*приводим столбцы fname и lname к нужному формату\*/

:new.fname:=

UPPER(substr(:new.fname,1,1))||LOWER(substr(:new.fname,2));

:new.lname:=

UPPER(substr(:new.lname,1,1))||LOWER(substr(:new.lname,2));

*end*;

Из рассмотренных выше примеров видно, что в PL/SQL имеются следующие уникальные языковые конструкции для триггеров баз данных:

- предикаты *inserting*, *updating*, *deleting* могут использоваться в операторах условного перехода в теле триггера.

- значения корреляции позволяют с помощью строковых триггеров обращаться к новым и старым значениям полей текущей строки. Когда оператором триггера является *insert* все старые значения полей являются *null* –значениями. Аналогично, когда оператором триггера является *delete* все новые значения являются соответственно null – значениями. Допустимы только для строчных триггеров.

Необходимо учитывать, что строчные триггеры не могут считывать или записывать данные из таблицы, с которой он связан. Но данное ограничение относится только к строчным триггерам, и не касается операторных триггеров.

При создании триггеров следует иметь в виду, что они принимают участие в транзакциях. Если в триггере инициируется исключение, то будет выполнен откат всей транзакции. Если триггер сам изменяет данные, то все изменения становятся частью общей транзакции. В триггере **нельзя применять** операторы COMMIT, ROLBACK и SAVEPOINT. Процедуры, вызываемые из триггера, также не могут использовать перечисленные выше операторы управления транзакциями, так как такая процедура исполняется внутри контекста тела триггера. Однако если триггер выполняется как автономная транзакция, т.е. он объявлен с инструкцией PRAGMA AUTONOMUS\_TRANSACTION , то использование в нём операторов управления транзакциями – обязательно и в нем можно запрашивать содержимое таблицы, но не изменять его.

**Варианты заданий**

**Вариант 1.**

1. Сотрудник не должен одновременно работать более чем на 1,5 ставки. Если при вставке или обновлении контракта размер ставки превышает 1,5, то операция не выполняется, а в сообщении указывается, сколько составляет превышение.
2. Руководить кафедрой может только один сотрудник с научной степенью, а количество сотрудников кафедры не должно превышать максимально возможного, которое задается при описании кафедры.
3. Отслеживать за месяц срок окончания контрактов сотрудников.

**Вариант 2.**

1) Реализовать политику премирования продавцов, в зависимости от количества или суммы продаж в месяц.

2)Управлять количеством товара в наличии при осуществлении продажи (вычитать количество купленного, запрещать продажи с недостающим количеством или предлагать купить остаток и т.д.), рассчитывать общую стоимость каждой покупки с учетом наценки магазина.

3)В определенное время сообщать о наличии продавцов, которые не продали за день (неделю, месяц) ни одного товара.

**Вариант 3.**

1) Осуществлять контроль загруженности почтальонов, например, по количеству одновременно обслуживаемых подписчиков.

2) Осуществлять расчет стоимости подписки, при этом предусматривать, чтобы нельзя было подписаться менее чем на месяц и более чем на год, а дата оформления подписки должна быть не позднее 10 дней до ее начала.

3) Каждый месяц перемещать подписки с законченным сроком в архив.

**Вариант 4.**

1) Расформировать отдел, перевести продавцов в другие так, чтобы среднее количество продавцов в отделе было приблизительно одинаковым.

2) Отслеживать наличие товара в необходимом количестве, изменять это количество при покупке или возврате, рассчитывать общую сумму продажи.

3) Осуществлять «скидочную» политику на товары, которые не были проданы в течение месяца (года). Сохранять во вспомогательной таблице старую цену и дату ее изменения.

**Вариант 5.**

1) Не превышать заданное количество сотрудников в отделах, предлагать (выводить) свободные места по другим отделам, а информацию о «лишних» сохранять во вспомогательной таблице.

2) Не допускать в штатном расписании работы сотрудников более чем на 1,5 ставки одновременно.

3) Начислять на определенную дату заработную плату всем работающим сотрудникам.

**Вариант 6.**

1) Реализовать ценовую политику при продаже пиломатериалов в зависимости от их количества, рассчитывать общую стоимость продажи с учетом количества товаров и стоимости сырья.

2) При недостаточном количестве сырья или изделий сохранять информацию о несостоявшейся продаже во вспомогательной таблице «Отложенная продажа».

3) Автоматически пополнять наличие товара на определенное количество раз в неделю и проверять возможность выполнения отложенных продаж.

**Вариант 7.**

1) Контролировать расход материалов при создании деталей, не создавать запись в технологической карте при недостаточном количестве необходимого материала.

2) Увольнять рабочего, только если все процессы с его участием завершены, переносить их во вспомогательную таблицу, указывая ФИО работника и дату его увольнения.

3) В определенный день составлять список материалов, которые заканчиваются (например, количество меньше, чем нужно на производство минимальной партии деталей).

**Вариант 8.**

1) Отслеживать количество пациентов, закрепленных за врачом одновременно, не допускать превышения установленного значения, которое зависит от категории и стажа специалиста.

2) При госпитализации контролировать занятость койко-мест, тип палат (мужская, женская) и срок нахождения пациента в больнице (не более 14 дней).

3) Каждый день обновлять информацию о количестве пациентов в отделении и количестве свободных мест по палатам.

**Вариант 9.**

1) Отслеживать наличие руководителя и количество человек в бригаде, так, чтобы оно было меньше заданного в триггере минимального значения и не превышало заданный максимум.

2) На одной единице оборудования в день не должно производиться больше технологических процессов, чем это установлено в таблице с описанием оборудования.

3) Обновлять максимальное и минимальное количество партий деталей, произведенных в день (месяц).

**Вариант 10.**

1) Отслеживать количество единиц товаров на складе при совершении оптовых продаж. Выдавать сообщение о необходимости пополнения склада, если оставшееся количество товара меньше, чем его среднее значение по продажам за последний месяц.

2) Организовать слежение за минимальным количеством опта товара в зависимости от его вида и производителя.

3) Создать таблицу-отчет, которая должна будет обновляться раз в неделю или раз в месяц и содержать данные о производителе, количестве проданных единиц товаров, сумме продаж.

**Вариант 11.**

1. Отслеживать загруженность оборудования по часам в актах работ, не допускать одновременную работу на одном оборудовании, работу в ночные часы и работу без перерывов.
2. Руководитель исследовательских работ может быть закреплен не более чем за тремя работами в месяц.
3. Архивировать информацию об оборудовании, подлежащем списанию во вспомогательную таблицу, где указывать общее количество актов, количество отработанных часов, дату начала и конца работ на каждой единице оборудования.

**Вариант 12.**

1. Рассчитывать общую сумму стоимости по договору, делая скидку клиентам в зависимости от количества или общей суммы их заказов.
2. Контролировать количество материала и продукции в наличии при заключении договора, не допускать одновременного оказания одних и тех же услуг нескольким клиентам.
3. Ежедневно вести учет договоров, по которым истечение сроков исполнения произойдет менее чем через три дня.

**Вариант 13.**

1. Рассчитывать сумму, которая должна быть уплачена риелтору в зависимости от типа сделки и включать ее в общую сумму оплаты.
2. Помечать специальным символом недвижимость, по которой была совершена сделка, или переносить ее из основной таблицы в архив.
3. Каждый день обновлять таблицу-отчет, содержащую тип недвижимости, тип сделки, количество объектов, сумму сделок, дату.

**Вариант 14.**

1. Автоматически формировать дату окончания обслуживания клиента в соответствии с гарантийными сроками на выбранное ПО; не допускать записей о гарантийных ремонтах после окончания гарантийного обслуживания.
2. При удалении поставщика информация о нем переносится в архив вместе с описанием его ПО и количеством заключенных сделок на покупку. Если есть сделки, по которым на момент удаления не завершено гарантийное обслуживание, удаление запретить.
3. Регулярно обновлять список постоянных клиентов (статус присваивается при покупке товаров на определенную сумму и аннулируется, если в течение года нет обращений).

**Вариант 15.**

1. При оформлении технологической карты контролировать рабочее время понедельник-пятница с 8.00 до17.00 (обед с 12.30 до 13.30) для оборудования и для рабочих, не допускать одновременной работы (график можно формировать самостоятельно, но не более 40 часов в неделю для каждого рабочего).
2. Следить за количеством инструмента на складе, его износом и соответствием определенному оборудованию при формировании карт.
3. Пополнять склад инструментов, если его количество достигает установленных минимальных значений.

**Вариант 16.**

1. При заполнении информации о договоре автоматически вносить текущую дату как начальную, если она не указана агентом, и дату окончания страховки, в зависимости от типа и срока страхования, рассчитывать общую стоимость и первоначальный взнос.
2. Следить за суммами выплат, начислять пеню за просроченные выплаты.
3. Переносить в архив старые договора, указывая сумму страховки и выплаты по страховым случаям, если они были.

**Вариант 17.**

1. Техника, находящаяся в ремонте, не должна быть задействована в других работах; на одной единице техники нельзя работать одновременно.
2. Предусмотреть, чтобы каждая единица техники проходила плановый ремонт в соответствии с установленным графиком.
3. Обновлять информацию во вспомогательной таблице о часах работы, планового и внепланового ремонта по каждой единице оборудования.

**Вариант 18**.

1. Запретить выставлять один экспонат на две выставки одновременно; выставлять экспонат, находящийся на реставрации.
2. Контролировать количество экспонатов на каждой выставке.
3. Если выставка закончилась, то во вспомогательной таблице обновлять «коэффициент выставляемости» по каждому экспонату и по каждому автору.

**Вариант 19.**

1. Рассчитывать заработную плату сотрудников в зависимости от оклада и процента участия в разработках.
2. Сотрудник одновременно работает не более чем на 1,5 ставки и может совмещать не более трех должностей, он не может работать одновременно более чем над 3 разработками и не более чем в 2 отделах.
3. Во вспомогательную таблицу вносить информацию о ведущихся в отделе разработках с указанием количества участвующих сотрудников, датах начала и окончания.

**Вариант 20**

1. В каждом фильме должен быть режиссер и, хотя бы один актер, при этом они не могут сниматься одновременно более чем в двух фильмах.
2. Предусмотреть расчет рейтинговых баллов актеров в зависимости от количества или сумм продаж фильма.
3. Архивировать продажи фильмов, которые не возобновлялись в течение года.

**Вариант 21.**

1. При формировании дат сдачи сессии контролировать занятость студентов и преподавателей, количество дней между экзаменами и количество экзаменов в сессию.
2. Проверять сданы ли студентом необходимые зачеты; не допускать студента до следующего экзамена, если количество неудовлетворительных оценок в текущую сессию достигло двух.
3. После сдачи последнего экзамена изменить в дополнительной таблице статус студента: переведен (не переведён) на курс №\_\_ и установить % повышения (понижения) стипендии.

**Вариант 22.**

1. При заполнении накладной устанавливать текущую дату, рассчитывать общую стоимость, проверять количество топлива в наличии.
2. Пополнять количество каждого вида топлива на складе, но не превышать максимально возможного общего количества на станции.
3. Заносить в дополнительную таблицу сведения об объеме топлива и общей сумме по каждому поставщику.

**Вариант 23.**

1. При вставке данных в таблицу «Организация выставки» проверять не занято ли уже выставочное место, запрещать отдавать более 20% мест на выставке одному предприятию.
2. Контролировать даты проведения выставок: количество дней выставки не менее 3-х, время между выставками не менее 2-х дней и т.д.
3. В дополнительной таблице обновлять информацию о количестве выставок по месяцам.

**Вариант 24.**

1. При изменении цен на материалы пересчитывать стоимость услуг, для которых они необходимы.
2. В месяц не заключать более определенного количества контрактов, делать скидку заказчику при повторном обращении, рассчитывать общую стоимость контракта автоматически.
3. Вести в дополнительной таблице учет занятости оборудования по дням недели.

**Вариант 25.**

1. При заключении контракта проверять и изменять наличие медтехники на складе,
2. Рассчитывать стоимость контракта с учетом скидки для постоянных клиентов.
3. Вести общий учет количества проданных товаров по поставщикам.

**Вариант 26.**

1. При формировании наряда проверять занятость работников и проведение других мероприятий на участке в одно время.
2. Контролировать количество работников по должностям и их общее количество.
3. Рассчитывать заработную плату работников по итогам месяца.

**Вариант 27.**

1. Отслеживать последовательное выполнение этапов по каждому объекту строительства с учетом сроков на выполнение.
2. Контролировать наличие материалов, занятость бригад и сроки строительства; не допускать одновременного строительства более 3 объектов.
3. Если последний этап строительства завершен, то сохранить данные в дополнительной таблице «Построенные объекты», где указывать данные об объекте и количество дней, потраченных на строительство.

**Вариант 28.**

1) При составлении технологических карт контролировать занятость оборудования и рабочих, не загружать оборудование больше 8 часов в день.

2) В зависимости от количества изготавливаемых деталей устанавливать время на их изготовление в карте.

3) Вести подсчет деталей, изготавливаемых на каждой единице оборудования в месяц.

**Вариант 29.**

1. Вести учет рабочего времени в цеху (время начала и конца работы в соответствии с установленным графиком, для рабочего не более 8-12 часов в день).
2. При изготовлении деталей контролировать количество материалов и оснастки на складе, квалификацию рабочего.
3. Вести учет изготовленных деталей на складе.

**Вариант 30.**

1. Для выбранной конфигурации установить и контролировать минимальные требования к оборудованию при заключении контракта.
2. У одного исполнителя не должно быть одновременно два договора и не более 3 договоров в день всего.
3. При удалении клиента архивировать в дополнительную таблицу информацию о нем и статистику по договорам.

**Вариант 31.**

1. Контролировать количества сырья и продукции при заключении контракта.
2. Вести скидочную политику для постоянных клиентов фирмы.
3. Протоколировать количество проданного сырья и стройматериалов.

**Вариант 32.**

1. Рассчитывать цену обслуживания с учетом стоимости работы и медикаментов, делать скидку 10%, если хозяйство обслуживается чаще 2 раз в месяц.
2. Контролировать расход медикаментов на проведенные работы.
3. В конце месяца автоматически формировать в проведенных работах записи о профилактических осмотрах обслуживаемых хозяйств.

**Вариант 33.**

1. Одна бригада работает одновременно только с одной скважиной.
2. Контролировать по параметрам соответствие скважины и оборудования для ее бурения при заключении контракта.
3. Списать оборудование и заархивировать общий метраж пробуренных скважин.

**Вариант 34.**

1. Контролировать занятость аудиторий, группы и преподавателей при любом изменении в таблице загрузки аудиторий.
2. Разрешить назначать пару только с 8.00 до 22.00 и учитывать соответствие типа аудиторий и проводимых занятий.
3. Считать общее количество часов занятий по кафедрам за семестр в дополнительной таблице.

**Вариант 35.**

1. При создании договора на сбыт проверять наличие изделий на складе. Если изделий не хватает, то сообщать об этом и разрешать вставку только при условии, что дата поставки не раньше, чем через две недели от текущей даты. Если изделий достаточно, то изменять количество товара на складе.
2. Организовать систему скидок для постоянных клиентов при расчете стоимости.
3. В дополнительной таблице обновлять информацию об изменениях в ассортименте фабрики.

**Вариант 36.**

1. Рассчитывать стоимость квитанции и контролировать расход материалов на заказанные изделия.
2. Не брать заказов больше, чем можно выполнить в ателье за месяц с учетом времени пошива на изделие.
3. Вести статистику о количестве заказов по кварталам.

**Вариант 37.**

1. Контролировать занятость номера и не заселять нового клиента в день выселения предыдущего; рассчитывать стоимость проживания.
2. За одним представителем персонала не может быть закреплено больше 5 номеров.
3. В дополнительной таблице фиксировать количество проживающих в каждом номере и общую сумму оплаты за месяц.

**Вариант 38.**

1. Контролировать соответствие мероприятия и квалификации специалиста, а также зону обслуживания.
2. При оформлении заказа следить, чтобы работник был свободен на дату исполнения, если он занят, то выводить сообщение о дате, когда он свободен.
3. При увольнении сотрудника разрешать переносить его в архив только с датой не ранее завершения последнего заказа.

**Вариант 39.**

1. Запрещать запись клиентов в одно и то же время к одному и тому же врачу,
2. Предусмотреть систему изменения цен на услуги в рамках заданного интервала значений и времени.
3. Уволить врача, учитывая дату последнего обслуживания и сохранить информацию о количестве обслуженных пациентов, количестве оказанных услуг и общей сумме оплат.

**Вариант 40.**

1. Контролировать последовательность и количество операций для сборки любого изделия.
2. Следить за наличием комплектующих и не собирать одновременно более 3 изделий.
3. Фиксировать время обновления количества комплектующих в дополнительной таблице.

**Вариант 41.**

1. При заключении контракта контролировать количество изделий и материалов в наличии, сроки изготовления, рассчитывать цену контракта.
2. Контролировать возможность изготовления изделий только из определенных материалов.
3. Если материал отсутствует на складе, то в дополнительной таблице записывать заказ на него, дату, требуемое количество и возможного поставщика.

**Вариант 42**.

1. При производстве контролировать наличие материалов, занятость рабочих и даты изготовления.
2. Контролировать соответствие технологических процессов, их количества и последовательности определенным видам кабеля.
3. Записывать во вспомогательную таблицу данные о произведенной продукции за день.

**Вариант 43.**

1. При формировании путевки контролировать занятость техники и водителей, делать перерывы между работами по установленному графику.
2. График работы водителей должен быть с 8.00 до 18.00 по схеме «два через два» и на определенной технике.
3. Контролировать расход топлива и километраж, пройденный каждой единицей техники в дополнительной таблице.

**Вариант 44.**

1. Контролировать количество одновременно отдыхающих в соответствии с количеством мест в санатории, контролировать количество отдыхающих, закрепленных за одним врачом.
2. Создать систему соответствий и противопоказаний между заболеваниями и оздоровительными программами, следить за правильностью назначений.
3. В отдельной таблице обновлять сведения о количестве дней, приведенных в санатории каждым отдыхающим.

**Вариант 45.**

1. Контролировать, чтобы в экспедиции было не менее 3 геологов, и все они не были задействованы в других экспедициях.
2. Подбирать все карты для выбранного маршрута.
3. При отсутствии свободных геологов данные о предполагаемой экспедиции заносить во вспомогательную таблицу.

**Вариант 46.**

1. Делать скидку клиенту при повторном обращении.
2. При составлении договора контролировать занятость персонала, стоимость работ, даты оказания услуги.
3. Ежедневно переносить во вспомогательную таблицу записи о договорах, которые должны быть завершены на текущей неделе.