

Глава 3

ЗАДАЧИ ПО РАЗРАБОТКЕ ПРОСТЫХ БАЗ ДАННЫХ

В этом разделе представлены варианты заданий для выполнения лабораторных работ. В каждом варианте задания описаны требования, предъявляемые к проектируемой базе данных.

В конце раздела приведен пример выполнения лабораторной работы.

3.1. Варианты заданий

1. **База данных хроники восхождений в альпинистском клубе.** В базе данных должны записываться даты начала и завершения каждого восхождения, имена и адреса участвовавших в нем альпинистов, название и высота горы, страна и район, где эта гора расположена. Дайте выразительные имена таблицам и полям, в которые могла бы заноситься указанная информация. Написать пакет, состоящий из процедур и функций, которые позволил бы выполнить следующие действия с базой данных:
 - 1) для каждой горы показать список групп, осуществлявших восхождение, в хронологическом порядке;
 - 2) предоставить возможность добавления новой вершины, с указанием названия вершины, высоты и страны местоположения;
 - 3) предоставить возможность изменения данных о вершине, если на нее не было восхождения;
 - 4) показать список альпинистов, осуществлявших восхождение в указанный интервал дат;

3.1. Варианты заданий

- 5) предоставить возможность добавления нового альпиниста в состав указанной группы;
- 6) показать информацию о количестве восхождений каждого альпиниста на каждую гору;
- 7) показать список восхождений (групп), которые осуществлялись в указанный пользователем период времени;
- 8) предоставить возможность добавления новой группы, указав ее название, вершину, время начала восхождения;
- 9) предоставить информацию о том, сколько альпинистов побывали на каждой горе.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

2. **База данных медицинского кооператива.** Базу данных использует для работы коллектив врачей. В таблицы должны быть занесены имя, пол, дата рождения и домашний адрес каждого их пациента. Всякий раз, когда врач осматривает больного, явившегося к нему на прием, или сам приходит к нему на дом, он записывает дату и место, где проводится осмотр, симптомы, диагноз и предписания больному, проставляет имя пациента, а также свое имя. Если врач прописывает больному какое-либо лекарство, в таблицу заносится название лекарства, способ его приема, словесное описание предполагаемого действия и возможных побочных эффектов. Создать пакет, состоящий из функций и процедур, позволяющих:

- 1) по заданной дате определить количество вызовов в этот день;
- 2) позволяют определить количество больных, заболевших данной болезнью;
- 3) по заданному лекарству определить его побочный эффект **порядке**;
- 4) предоставить возможность добавления нового лекарства с описанием его свойств в БД.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

3. **База данных Городской Думы.** В базе хранятся имена, адреса, домашние и служебные телефоны всех членов Думы. В Думе работает порядка сорока комиссий, все участники которых являются членами Думы. Каждая комиссия имеет свой профиль, например, вопросы образования, проблемы, связанные с жильем и так далее. Данные по каждой из комиссий включают: ее нынешний состав и председатель, прежние председатели и члены этой комиссии, участвовавшие в ее работе за прошедшие 10 лет, даты включения и выхода из состава комиссии, избрания ее председателей. Члены Думы могут заседать в нескольких комиссиях. В базу заносятся время и место проведения каждого заседания комиссии с указанием депутатов и служащих Думы, которые участвуют в его организации. Создать пакет с процедурами и функциями, которые позволяют выполнять следующие действия:

- 1) показать список комиссий, для каждой ее состав и председателя;
- 2) предоставить возможность добавления нового члена комиссии;
- 3) показать список членов муниципалитета, для каждого из них список комиссий, в которых он участвовал и/или был председателем;
- 4) предоставить возможность добавления новой комиссии, с указанием председателя;
- 5) для указанного интервала дат и комиссии выдать список ее членов с указанием количества пропущенных заседаний;
- 6) предоставить возможность добавления нового заседания, с указанием присутствующих;
- 7) по каждой комиссии показать количество проведенных заседаний в указанный период времени.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

4. **База данных рыболовной фирмы.** Фирме принадлежит небольшая флотилия рыболовных катеров. Каждый катер имеет

3.1. Варианты заданий

“паспорт”, куда занесены его название, тип, водоизмещение и дата постройки. Фирма регистрирует каждый выход на лов, записывая название катера, имена и адреса членов команды с указанием их должностей (капитан, боцман и т.д.), даты выхода и возвращения, а также вес пойманной рыбы отдельно по сортам (например, трески). За время одного рейса катер может посетить несколько банок. Фиксируется дата прихода на каждую банку и дата отплытия, качество выловленной рыбы (отличное, хорошее, плохое). На борту улов не взвешивается. Написать запросы, осуществляющие следующие операции:

- 1) для каждого катера вывести даты выхода в море с указанием улова;
- 2) предоставить возможность добавления выхода катера в море с указанием команды;
- 3) для указанного интервала дат вывести для каждого сорта рыбы список катеров с наибольшим уловом;
- 4) для указанного интервала дат вывести список банок, с указанием среднего улова за этот период;
- 5) предоставить возможность добавления новой банки с указанием данных о ней;
- 6) для заданной банки вывести список катеров, которые получили улов выше среднего;
- 7) вывести список сортов рыбы и для каждого сорта список рейсов с указанием даты выхода и возвращения, количества улова;
- 8) для выбранного пользователем рейса и банки добавить данные о сорте и количестве пойманной рыбы;
- 9) предоставить возможность пользователю изменять характеристики выбранного катера;
- 10) предоставить возможность добавления нового катера;
- 11) для указанного сорта рыбы и банки вывести список рейсов с указанием количества пойманной рыбы.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

5. **База данных фирмы, проводящей аукционы.** Фирма занимается продажей с аукциона антикварных изделий и произведений искусства. Владельцы вещей, выставяемых на проводимых фирмой аукционах, юридически являются продавцами. Лица, приобретающие эти вещи, именуются покупателями. Получив от продавцов партию предметов, фирма решает, на котором из аукционов выгоднее представить конкретный предмет. Перед проведением очередного аукциона каждой из выставяемых на нем вещей присваивается отдельный номер лота, играющий ту же роль, что и введенный ранее шифр товара. Две вещи, продаваемые на различных аукционах, могут иметь одинаковые номера лотов.

В книгах фирмы делается запись о каждом аукционе. Там отмечаются дата, место и время его проведения, а также специфика (например, выставяются картины, написанные маслом и не ранее 1900 г.). Заносятся также сведения о каждом продаваемом предмете: аукцион, на который он заявлен, номер лота, продавец, отправная цена и краткое словесное описание. Продавцу разрешается выставять любое количество вещей, а покупатель имеет право приобретать любое количество вещей. Одно и то же лицо или фирма может выступать и как продавец, и как покупатель. После аукциона служащие фирмы, проводящей аукционы, записывают фактическую цену, уплаченную за проданный предмет, и фиксируют данные покупателя.

Создать пакет, состоящий из процедур и функций, позволяющий осуществить следующие операции:

- 1) для указанного интервала дат вывести список аукционов с указанием наименования, даты и места проведения;
- 2) добавить на указанный пользователем аукцион на продажу предмет искусства с указанием начальной цены;
- 3) вывести список аукционов, с указанием суммарного дохода от продажи, отсортированных по доходу;

3.1. Варианты заданий

- 4) для указанного интервала дат, вывести список предметов, которые были проданы на аукционах в этот период времени;
- 5) предоставить возможность добавления факта продажи на указанном аукционе заданного предмета;
- 6) для указанного интервала дат вывести список продавцов с указанием общей суммы, полученной от продажи предметов в этот промежуток времени;
- 7) вывести список покупателей, которые сделали приобретения в указанный интервал дат;
- 8) предоставить возможность добавления записи о проводимом аукционе (место, время);
- 9) для указанного места, вывести список аукционов;
- 10) для указанного интервала дат вывести список продавцов, которые принимали участие в аукционах, проводимых в этот период времени;
- 11) предоставить возможность добавления и изменения информации о продавцах и покупателях;
- 12) вывести список покупателей с указанием количества приобретенных предметов в указанный период времени.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

6. **База данных библиотеки.** Разработать информационную систему обслуживания библиотеки, которая содержит следующую информацию: названия книг, ФИО авторов, наименования издательств, год издания, количество страниц, количество иллюстраций, стоимость, название филиала библиотеки или книгохранилища, в которых находится книга, количество имеющихся в библиотеке экземпляров конкретной книги, количество студентов, которым выдавалась конкретная книга, названия факультетов, в учебном процессе которых используется указанная книга. Необходимо составить пакет из процедур и функций, который позволяет:

- 1) для указанного филиала посчитать количество экземпляров указанной книги, находящихся в нем;
- 2) для указанной книги посчитать количество факультетов, на которых она используется в данном филиале, и вывести названия этих факультетов;
- 3) предоставить возможность добавления и изменения информации о книгах в библиотеке;
- 4) предоставить возможность добавления и изменения информации о филиалах;
- 5) предусмотреть разработку триггеров, срабатывающих на пользовательские исключительные ситуации;

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

7. **База данных по учету успеваемости студентов.** База данных должна содержать данные о контингенте студентов (фамилия, имя, отчество, год поступления, форма обучения (дневная/вечерняя/заочная), номер или название группы); об учебном плане (название специальности, дисциплина, семестр, количество отводимых на дисциплину часов, форма отчетности (экзамен/зачет)); о журнале успеваемости студентов (год/семестр, студент, дисциплина, оценка). Разработать пакет, состоящий из процедур и функций, позволяющий:

- 1) для указанной формы обучения посчитать количество студентов этой формы обучения;
- 2) для указанной дисциплины получить количество часов и формы отчетности по этой дисциплине; предоставить возможность добавления и изменения информации о студентах, об учебных планах, о журнале успеваемости при этом предусмотреть курсоры, срабатывающие на некоторые пользовательские исключительные ситуации;
- 3) предоставить возможность добавления и изменения информации о журнале успеваемости.

3.1. Варианты заданий

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

8. **База данных для учета аудиторного фонда университета.** База данных должна содержать следующую информацию об аудиторном фонде университета. Наименование корпуса, в котором расположено помещение, номер комнаты, расположение комнаты в корпусе, ширина и длина комнаты в метрах, назначение и вид помещения, подразделение университета, за которым закреплено помещение. В базе данных также должна быть информация о высоте потолков в помещениях в зависимости от места расположения помещений в корпусе. Следует также учитывать, что структура подразделений университета имеет иерархический вид, когда одни подразделения входят в состав других (факультет, кафедра, лаборатория).

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий:

- 1) рассчитать данные о площадях и объемах каждого помещения;
- 2) для указанного корпуса получить количество факультетов, их названия и структуру, находящиеся в этом корпусе;
- 3) предоставить возможность добавления и изменения информации о корпусах в университете, при этом предусмотреть курсоры, срабатывающие на некоторые пользовательские исключительные ситуации;
- 4) предоставить возможность добавления и изменения информации о комнатах в корпусах университета, при этом предусмотреть курсоры, срабатывающие на некоторые пользовательские исключительные ситуации.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

9. **База данных для регистрации происшествий.** Необходимо создать Базу данных для регистрации происшествий. База дан-

ных должна содержать данные для регистрации сообщений о происшествиях (регистрационный номер сообщения, дата регистрации, краткая фабула (тип происшествия); информацию о принятом по происшествию решении (отказано в возбуждении дел, удовлетворено ходатайство о возбуждении уголовного дела с указанием регистрационный номера заведенного дела, отправлено по территориальному признаку); информацию о лицах, виновных или подозреваемых в совершении происшествия (регистрационный номер лица, фамилия, имя, отчество, адрес, количество судимостей), отношение конкретных лиц к конкретным происшествиям (виновник, потерпевший, подозреваемый, свидетель...):

- 1) помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий;
- 2) рассчитать данные о количестве происшествий в указанный промежуток времени;
- 3) для указанного лица получить количество происшествий, в которых он зарегистрирован;
- 4) предоставить возможность добавления и изменения информации о происшествиях, при этом предусмотреть курсоры, срабатывающие на некоторые пользовательские исключительные ситуации;
- 5) предоставить возможность добавления и изменения информации о лицах, участвующих в происшествиях, при этом предусмотреть курсоры, срабатывающие на некоторые пользовательские исключительные ситуации.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

10. **База данных для обслуживания работы конференции.** База данных должна содержать справочник персоналий участников конференции (фамилия, имя, отчество, ученая степень, ученое звание, научное направление, место работы, кафедра (отдел), должность, страна, город, почтовый индекс, адрес, рабочий те-

3.1. Варианты заданий

елефон, домашний телефон, e-mail), и информацию, связанную с участием в конференции (докладчик или участник, дата рассылки 1-го приглашения, дата поступления заявки, тема доклада, отметка о поступлении тезисов, дата рассылки 2-го приглашения, дата поступления оргвзноса, размер поступившего оргвзноса, дата приезда, дата отъезда, потребность в гостинице).

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий:

- 1) для указанной даты 1-ой рассылки вывести список приглашенных и посчитать их количество;
- 2) предоставить возможность добавления приглашенных на конференцию с указанием оргвзноса и даты его уплаты;
- 3) вывести список приглашенных, с указанием даты об уплате оргвзноса;
- 4) для указанной интервала дат, вывести список участников, уплативших оргвзнос в этом диапазоне;
- 5) для указанного города вывести название тезисов докладов, поступивших из этого города;
- 6) для указанного города, вывести список нуждающихся в гостинице.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

11. База данных для обслуживания склада. База данных должна обеспечить автоматизацию складского учета. В ней должны содержаться следующие данные:

- информация о “единицах хранения” — номер ордера, дата, код поставщика, балансный счет, код сопроводительного документа по справочнику документов, номер сопроводительного документа, код материала по справочнику материалов, счет материала, код единицы измерения, количество пришедшего материала, цена единицы измерения);

- информация о хранящихся на складе материалах (справочник материалов — код класса материала, код группы материала, наименование материала);
- информация о единицах измерения конкретных видов материалов — код материала, единица измерения (метры, килограммы, литры и т.д.);
- информация о поставщиках материалов — код поставщика, его наименование, ИНН, юридический адрес (индекс, город, улица, дом), адрес банка (индекс, город, улица, дом), номер банковского счета.

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий:

- 1) посчитать количество поставщиков данного материала;
- 2) предоставить возможность добавления единицы хранения с указанием всех реквизитов;
- 3) вывести список поставщиков с указанием всех реквизитов данного материала на склад;
- 4) для указанного адреса банка посчитать количество поставщиков склада, пользующихся услугами этого банка.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

12. **База данных фирмы.** Фирма отказалась от приобретения некоторых товаров у своих поставщиков, решив самостоятельно наладить их производство. С этой целью она организовала сеть специализированных цехов, каждый из которых принимает определенное участие в технологическом процессе.

Каждому виду выпускаемой продукции присваивается, как обычно, свой шифр товара, под которым он значится в файле товарных запасов. Этот же номер служит и шифром продукта. В записи с этим шифром указывается, когда была изготовлена последняя партия этого продукта, какова ее стоимость, сколько операций потребовалось.

3.1. Варианты заданий

Операцией считается законченная часть процесса производства, которая целиком выполняется силами одного цеха в соответствии с техническими требованиями, перечисленными на отдельном чертеже. Для каждого продукта и для каждой операции в базе данных фирмы заведена запись, содержащая описание операции, ее среднюю продолжительность и номер чертежа, по которому можно отыскать требуемый чертеж. Кроме того, указывается номер цеха, обычно производящего данную операцию.

В запись, связанную с конкретной операцией, заносятся потребные количества расходуемых материалов, а также присвоенные им шифры товара. Расходуемыми называют такие материалы, как, например, электрический кабель, который нельзя использовать повторно. Когда, готовясь к выполнению операции, расходуемый материал забирают со склада, регистрируется фактически выданное количество, соответствующий шифр товара, номер служащего, ответственного за выдачу, дата и время выдачи, номер операции и номер наряда на проведение работ, который будет обсуждаться ниже. Реально затраченное количество материала может не совпадать с расчетным, из-за того, например, что часть изготовленной продукции бракуется.

Каждый из цехов располагает многочисленными инструментами и приспособлениями. При выполнении некоторых операций их все же не хватает, и цех вынужден обращаться в центральную инструментальную за недостающими. Каждый тип инструмента снабжен отдельным номером и на него заведена запись со словесным описанием. Кроме того, там отмечено, какое количество инструментов этого типа выделено цехам и какое осталось в инструментальной. Экземпляры инструмента конкретного типа, например гаечные ключи одного размера, различаются по своим индивидуальным номерам. На фирме для каждого типа инструмента имеется запись, содержащая перечень всех индивидуальных номеров. Кроме того, указаны даты их поступления на склад. По каждой операции в фирме отмечают типы и количества инструментов этих типов, которые должны использоваться при ее выполнении. Когда инструменты действительно берутся со скла-

да, фиксируется индивидуальный номер каждого экземпляра, указываются номер заказавшего их цеха и номер наряда на проведение работ. И в этом случае затребованное количество не всегда совпадает с заказанным.

Наряд на проведение работ по форме напоминает заказ на приобретение товаров, но, в отличие от последнего, он направляется не поставщику, а в один из цехов. Оформляется этот наряд после того, как руководство фирмы сочтет необходимым выпустить партию некоторого продукта. В наряд заносятся шифр продукта, дата оформления наряда, срок, к которому должен быть выполнен заказ, а также требуемое количество.

Задача 1. Разработайте структуру таблиц базы данных, подберите имена таблиц и полей, в которых могла бы разместиться вся эта информация.

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий:

- 1) для выбранного цеха, выдать список операций, выполняемых им. Для каждой операции список расходуемых материалов, с указанием количества;
- 2) показать список инструментов и предоставить возможность добавления нового;
- 3) выдать список используемых инструментов;
- 4) для указанного интервала дат, вывести список нарядов;
- 5) показать список операций и предоставить возможность добавления новой операции;
- 6) выдать список расходуемых материалов, используемых в различных нарядах;
- 7) выдать список товаров, с указанием используемых инструментов;
- 8) показать список нарядов и предоставить возможность добавления нового;

3.1. Варианты заданий

- 9) выдать отчет о производстве товаров различными цехами, указав наименование цеха, название товара и его количество.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

- 13. База данных музыкального магазина.** Таблицы базы данных содержат информацию о музыкантах, музыкальных произведениях и обстоятельствах их исполнения. Несколько музыкантов, образующих единый коллектив, называются ансамблем. Это может быть классический оркестр, джазовая группа, квартет, квинтет и т.д. К музыкантам причисляют исполнителей (играющих на одном или нескольких инструментах), композиторов, дирижеров и руководителей ансамблей.

Кроме того, в базе данных хранится информация о пластинках, которыми магазин торгует. Каждая пластинка, а точнее, ее наклейка, идентифицируется отдельным номером, так что всем копиям, отпечатанным с матрицы в разное время, присвоены одинаковые номера. На пластинке может быть записано несколько исполнений одного и того же произведения — для каждого из них в базе заведена отдельная запись. Когда выходит новая пластинка, регистрируется название выпускавшей ее компании (например, ЕМП), а также адрес оптовой фирмы, у которой магазин может приобрести эту пластинку. Не исключено, что компания—производитель занимается и оптовой продажей своих пластинок. Магазин фиксирует текущие оптовые и розничные цены на каждую пластинку, дату ее выпуска, количество экземпляров, проданных за прошлый год и в нынешнем году, а также число еще не распроданных пластинок.

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющих:

- 1) количество музыкальных произведений заданного ансамбля;
- 2) выводит название всех компакт-дисков заданного ансамбля;

- 3) показать лидеров продаж текущего гола, то есть названия компакт-дисков, которые чаще всего покупали в текущем году;
- 4) предусмотреть изменения данных о компакт-дисках и ввод новых данных;
- 5) предусмотреть ввод новых данных об ансамблях.

Предусмотреть разработку триггеров, обеспечивающих каскадные изменения в связанных таблицах.

3.2. Пример выполнения лабораторной работы

3.2.1. Постановка задачи

База данных “Скачки”. В информационной системе клуба любителей скачек должна быть представлена информация об участвующих в скачках лошадях (кличка, пол, возраст), их владельцах (имя, адрес, телефон) и жокеях (имя, адрес, возраст, рейтинг). Необходимо сформировать таблицы для хранения информации по каждому состязанию: дата, время и место проведения скачек (ипподром), название состязаний (если таковое имеется), клички участвующих в заездах лошадей и имена жокеев, занятые ими места и показанное в заезде время.

Требуется:

- сформировать структуру таблиц базы данных;
- подобрать подходящие имена таблицам и их полям;
- обеспечить требования нормализации таблиц базы данных;
- сформировать SQL-запросы для создания таблиц базы данных с указанием первичных и внешних ключей и требуемых ограничений;
- создать пакет с курсором и триггеры.

3.2.2. Описание структуры таблиц базы данных

Опишем структуру таблиц, связи между ними и произведем их нормализацию.

Создадим две вспомогательные таблицы **Pol** и **Ippodrom**.

- Таблица **Pol** содержит возможные значения пола лошади (поле **Pol_Value**) и их порядковые номера (поле **Force_Pol**). Поле **Force_Pol** является первичным ключом таблицы.

Force_Pol	Pol_Value
1	‘мужской’
2	‘женский’

- Таблица **Ippodrom** содержит возможные названия ипподромов (поле **Ippodrom_Name**) и их порядковые номера (поле **Ippodrom_id**). Поле **Ippodrom_id** является первичным ключом таблицы.

Ippodrom_id	Ippodrom_Name
1	‘Чесменка’
2	‘Спортсмен’
3	‘Профессионал’
...

- Таблица **Forces** содержит информацию о лошадях:

Force_id	Force_Name	Force_Pol	Force_Age	Vladel_id
1	Ветерок	1	3	1
2	Стрела	2	2	4
.....

Эта таблица содержит следующие поля.

Force_id – содержит идентификатор лошади и это поле объявлено как первичный ключ.

Force_Name – содержит имя лошади и является уникальным.

Force_Pol – содержит информацию о поле лошади и является внешним ключом (оно ссылается на поле **Force_Pol** таблицы **Pol**).

Таблица Gokey

Force_Age – содержит информацию о возрасте лошади.

Vladel_id – содержит информацию о владельце лошади и является внешним ключом (ссылается на поле Vladel_id таблицы Vladel).

- Таблица **Gokey** содержит информацию о жокеях:

Gokey_id	Gokey_Name	Gokey_Address	Gokey_Age	Gokey_Reiting
1	Ивлев	Москва	21	314
2	Семенов	Москва	35	245
.....

Эта таблица содержит следующие поля.

Gokey_id – содержит идентификатор жокея и это поле объявлено как первичный ключ.

Gokey_Name – содержит имя жокея.

Gokey_Address – содержит адрес жокея.

Gokey_Age – содержит информацию о возрасте жокея и имеет ограничение ($\text{Gokey_Age} \geq 18$).

Gokey_Reiting – содержит информацию о рейтинге жокея и имеет ограничение ($\text{Gokey_Reiting} \geq 0$).

- Таблица **Vladel** содержит информацию о владельцах лошадей:

Vladel_id	Vladel_Name	Vladel_Address	Vladel_Telephon
1	Петров	Москва	1652763
2	Сидоров	СПБ	6525622
....

Эта таблица содержит следующие поля.

Vladel_id – содержит идентификатор владельца и это поле объявлено как первичный ключ.

Vladel_Name – содержит имя владельца.

Vladel_Address – содержит адрес владельца.

Vladel_Telephon – содержит телефон владельца.

- Таблица **Zabeg** содержит информацию о забегах:

3.2. Пример выполнения лабораторной работы

Zabeg_id	Zabeg_Data	Zabeg_Time	Ippodrom_id	Zabeg_Name
1	'1-APR-02'	'18:30'	1	'Aprelski'
2	'3-MAY-02'	'12:30'	3	'Mayski'
.....

Эта таблица содержит следующие поля.

Zabeg_id – содержит идентификатор забега и это поле объявлено как первичный ключ.

Zabeg_Data – содержит дату забега.

Zabeg_Time – содержит время забега.

Ippodrom_id – содержит идентификатор ипподрома и является внешним ключом (ссылается на поле Ippodrom_id таблицы **Ippodrom**).

Zabeg_Name – содержит информацию о названии забега.

- Таблица **Results_Zabeg** содержит результаты забегов:

Rez_id	Zabeg_id	Force_id	Gokey_id	Result	Rez_Time
1	1	1	2	1	12.23
2	1	2	1	2	12.45
.....

Эта таблица содержит следующие поля.

Rez_id – содержит идентификатор результата забега и это поле объявлено как первичный ключ.

Zabeg_id – содержит идентификатор забега и является внешним ключом (ссылается на поле Zabeg_id таблицы **Zabeg**).

Force_id – содержит идентификатор лошади и является внешним ключом (ссылается на поле Force_id таблицы **Forces**).

Gokey_id – содержит идентификатор жокея и является внешним ключом (ссылается на поле Gokey_id таблицы **Gokey**).

Result – содержит информацию о занятом месте.

Rez_Time – содержит информацию о показанном в заезде времени.

Концептуальная модель базы данных приведена на рис. 3.1.

Концептуальная модель базы данных

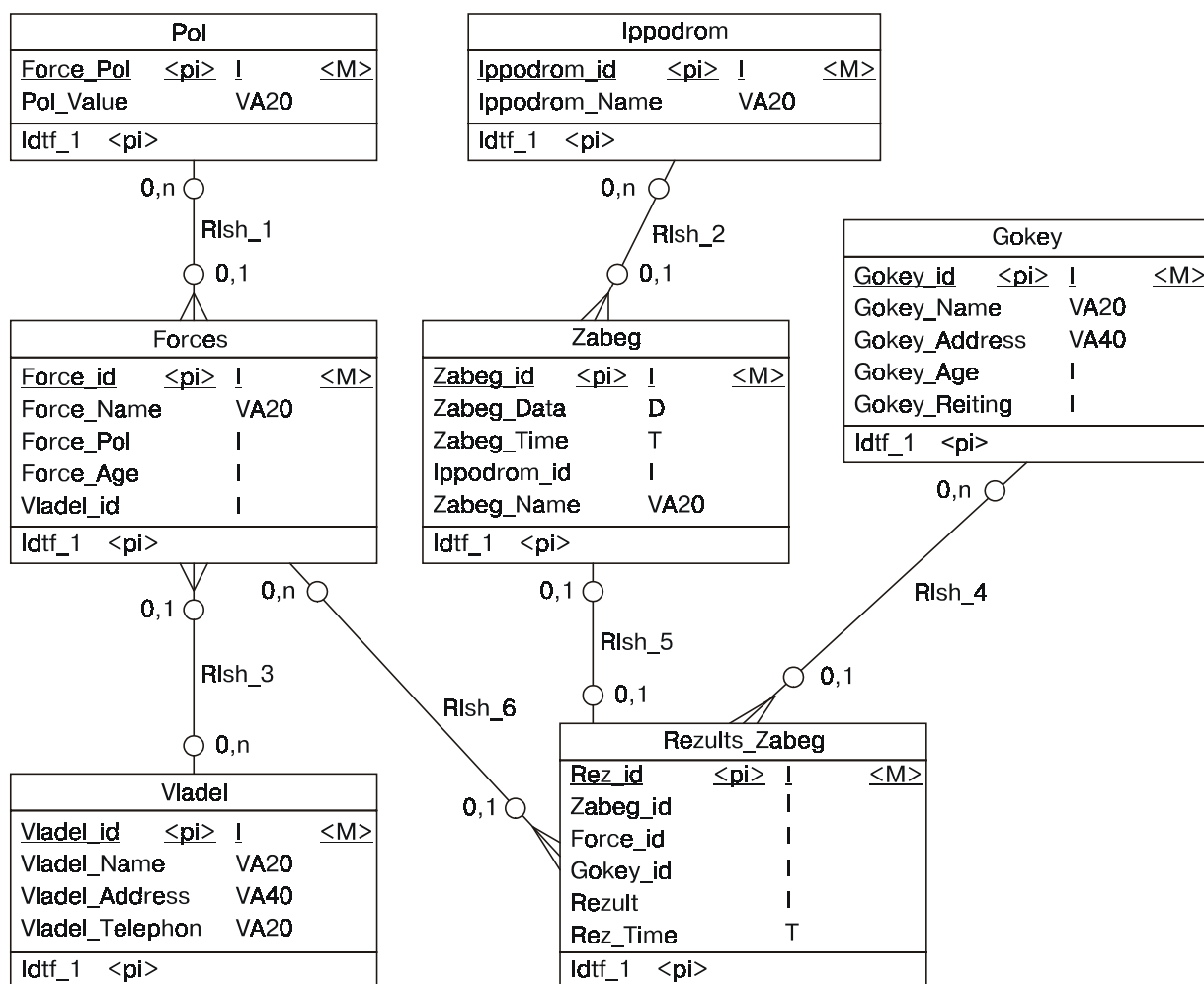


Рис. 3.1. Концептуальная схема базы данных

Во всех таблицах изменения родительского ключа разрешаются, но при этом осуществляется коррекция всех значений внешних ключей, ссылающихся на модифицируемое значение родительского ключа. Таким образом обеспечивается ссылочная целостность базы данных. База данных находится в нормализованном состоянии.

3.2.3. Создание и заполнение таблиц

Таблица “Пол”:

```
create table Pol
(force_pol      integer primary key,
 pol_value      varchar(10) not null);
```

3.2. Пример выполнения лабораторной работы

Заполнение таблицы:

```
insert into Pol values (1, 'man');  
insert into Pol values (2, 'woman');
```

Таблица “Ипподромы”:

```
create table ippodrom  
(ippodrom_id integer primary key,  
  ippodrom_name varchar(20) unique not null);
```

Заполнение таблицы:

```
insert into ippodrom values (1, 'Chesmenka');  
insert into ippodrom values (2, 'Sportsmen');  
insert into ippodrom values (3, 'Professional');
```

Таблица “Лошади”:

```
create table forces  
(force_id integer primary key,  
  force_name varchar(20) unique not null,  
  force_pol integer,  
  force_age integer not null,  
  vladel_id integer,  
  FOREIGN KEY (force_pol) REFERENCES pol,  
  FOREIGN KEY (vladel_id) REFERENCES vladel);
```

Заполнение таблицы:

```
insert into forces values (1, 'Veterok', 1, 3, 1);  
insert into forces values (2, 'Strela', 2, 2, 4);  
insert into forces values (3, 'Nochka', 2, 1, 3);  
insert into forces values (4, 'Pobeditel', 1, 2, 2);  
insert into forces values (5, 'Mig', 1, 1, 1);  
insert into forces values (6, 'Vasilinka', 2, 3, 2);
```

Таблица “Жокеи”:

```
create table gokey  
(gokey_id integer primary key,  
  gokey_name varchar(20) not null,  
  gokey_address varchar(40) not null,  
  gokey_age integer check (gokey_age >= 18),  
  gokey_reiting integer check (gokey_reiting >= 0));
```

Заполнение таблицы:

```
insert into gokey values
    (1, 'Ivlev', 'Moskva', 21, 314);
insert into gokey values
    (2, 'Semenov', 'Moskva', 35, 245);
insert into gokey values
    (3, 'Nikitin', 'Voronegh', 25, 450);
insert into gokey values
    (4, 'Shevelev', 'Moskva', 28, 190);
insert into gokey values
    (5, 'Popov', 'Krasnodar', 33, 254);
insert into gokey values
    (6, 'Krasnov', 'SPB', 23, 120);
```

Таблица “Владельцы”:

```
create table vladel
(vladel_id          integer primary key,
 vladel_name        varchar(20) not null,
 vladel_address     varchar(40) not null,
 vladel_telephon    varchar(20) not null);
```

Заполнение таблицы:

```
insert into vladel values
    (1, 'Petrov', 'Moskva', 1652763);
insert into vladel values
    (2, 'Sidorov', 'SPB', 6525622);
insert into vladel values
    (3, 'Semenov', 'Minsk', 7657652);
insert into vladel values
    (4, 'Krasnova', 'Samara', 8766632);
```

Таблица “Забег”:

```
create table zabeg
(zabeg_id integer primary key,
 zabeg_data date not null,
 zabeg_time varchar(20) not null,
 ippodrom_id integer,
 zabeg_name varchar(20),
 FOREIGN KEY (ippodrom_id) REFERENCES ippodrom);
```

3.2. Пример выполнения лабораторной работы

Заполнение таблицы:

```
insert into zabeg values
    (1, '1-APR-02', '18:30', 1, 'Aprelski');
insert into zabeg values
    (2, '3-MAY-02', '12:30', 3, 'Mayski');
insert into zabeg values
    (3, '1-SEP-02', '14:00', 2, 'Septem');
insert into zabeg values
    (4, '1-DEC-02', '18:30', 1, 'NewYear');
insert into zabeg values
    (5, '1-FEB-03', '18:30', 2, 'Febrary');
insert into zabeg values
    (6, '15-APR-03', '11:00', 3, 'Aprelski03');
insert into zabeg values
    (7, '1-OCT-03', '18:30', 1, 'October03');
```

Таблица “Результаты забегов”:

```
create table results_zabeg
(rez_id integer primary key,
 zabeg_id integer,
 force_id integer,
 gokey_id integer,
 result integer not null,
 rez_time real not null,
 FOREIGN KEY(force_id) REFERENCES forces,
 FOREIGN KEY(gokey_id) REFERENCES gokey,
 FOREIGN KEY(zabeg_id) REFERENCES zabeg);
```

Заполнение таблицы:

```
insert into results_zabeg values (1, 1, 1, 2, 1, 12.23);
insert into results_zabeg values (2, 1, 2, 1, 2, 12.45);
insert into results_zabeg values (3, 1, 4, 4, 3, 14.21);
insert into results_zabeg values (4, 1, 3, 3, 4, 15.23);
insert into results_zabeg values (5, 2, 1, 2, 3, 16.23);
insert into results_zabeg values (6, 2, 2, 1, 1, 12.43);
insert into results_zabeg values (7, 2, 3, 3, 2, 12.56);
insert into results_zabeg values (8, 3, 1, 1, 1, 12.23);
insert into results_zabeg values (9, 3, 2, 2, 2, 14.23);
insert into results_zabeg values (10, 3, 4, 3, 3, 15.03);
```

Функция Kolvo_Forces_Of_Vladel(Arg Forces.vladel_id%TYPE)

3.2.4. Описание программного продукта

Опишем процедуры и функции, которые вошли в пакет PASCZABEG, созданный на языке PL\SQL для работы с базой данных “Скачки”.

Функция

Kolvo_Forces_Of_Vladel(Arg Forces.vladel_id%TYPE)

возвращает натуральное значение, которое показывает количество лошадей, имеющих у заданного владельца (Arg – идентификатор владельца).

Процедура

Forces_Of_Vladel(Arg vladel.vladel_name%TYPE)

выводит список кличек лошадей, имеющих у заданного владельца (Arg – имя владельца). В процедуре используется курсор, содержащий запрос с выборкой, имеющий следующий вид:

```
Cursor CurForcel IS SELECT Forces.Force_name
FROM forces
WHERE vladel_id=(select vladel_id
from vladel where vladel_name=arg)
```

Процедура

Inform_About_Vladel(Arg Forces.Force_Name%TYPE)

выводит информацию о владельце (имя, адрес, телефон) по заданной кличке лошади (Arg). В процедуре используется курсор, содержащий запрос с выборкой, имеющий следующий вид:

```
Cursor CurVladell IS
SELECT vladel_name,
vladel_address,vladel_telephon
FROM Vladel
WHERE vladel_id=(Select vladel_id From Forces
WHERE force_name=Arg)
```

Процедура

Inform_About_Zabegs(Arg Zabeg.zabeg_data%TYPE)

выводит информацию о забеге (клички участвующих лошадей, имена жокеев, занятые места и показанное в забеге время) по задан-

3.2. Пример выполнения лабораторной работы

ной дате забега (Arg). В процедуре используется курсор, содержащий запрос с выборкой, имеющий следующий вид:

```
Cursor CurZabeg1 IS
SELECT Forces.force_name,Gokey.gokey_name,
       results_zabeg.result,results_zabeg.rez_time
FROM Forces,Gokey,results_zabeg
WHERE (Forces.force_id=results_zabeg.force_id)AND
       (Gokey.gokey_id=results_zabeg.gokey_id)AND
       results_zabeg.zabeg_id=(select zabeg_id
                                from zabeg where zabeg_data=arg)
```

Процедура **Liders_Zabeg** показывает лидеров всех забегов. В процедуре используется курсор, содержащий запрос с выборкой, имеющий следующий вид:

```
Cursor CurLider1 IS
SELECT Forces.Force_name,Gokey.gokey_name,
       results_zabeg.rez_time,zabeg.zabeg_data
FROM Forces,Gokey,results_zabeg,zabeg
WHERE (results_zabeg.result=1)AND
       (forces.force_id=results_zabeg.force_id)AND
       (gokey.gokey_id=results_zabeg.gokey_id)AND
       (zabeg.zabeg_id=results_zabeg.zabeg_id)
```

Процедура

```
New_Zabeg(Arg1 zabeg.zabeg_id%TYPE,
          Arg2 zabeg.zabeg_data%TYPE,
          Arg3 zabeg.zabeg_time%TYPE,
          Arg4 zabeg.ippodrom_id%TYPE,
          Arg5 zabeg.zabeg_name%TYPE,
          Arg6 results_zabeg.rez_id%TYPE,
          Arg7 results_zabeg.force_id%TYPE,
          Arg8 results_zabeg.gokey_id%TYPE,
          Arg9 results_zabeg.result%TYPE,
          Arg10 results_zabeg.rez_time%TYPE)
```

осуществляет ввод данных для нового забега. При этом происходит добавление в таблицах Zabeg и Results_Zabeg.

Создадим триггеры для обеспечения каскадных изменений в связанных таблицах базы данных “Скачки”.

- Триггер **TRIG_Forces_Pol** используется для таблицы **Pol** и обеспечивает каскадные изменения в связанных с ней таблицах.
- Триггер **TRIG_Forces_Vladel** используется для таблицы **Vladel** и обеспечивает каскадные изменения в связанных с ней таблицах.
- Триггер **TRIG_Zabeg_Ipp** используется для таблицы **Ippodrom** и обеспечивает каскадные изменения в связанных с ней таблицах.
- Триггер **TRIG_RezZabeg_Force** используется для таблицы **Forces** и обеспечивает каскадные изменения в связанных с ней таблицах.
- Триггер **TRIG_RezZabeg_Gokey** используется для таблицы **Gokey** и обеспечивает каскадные изменения в связанных с ней таблицах.

3.2.5. Создание пакета

Создание заголовка пакета:

```
CREATE OR REPLACE PACKAGE PACZABEG
AS
FUNCTION
Kolvo_Forces_Of_Vladel(Arg Forces.vladel_id%TYPE)
RETURN INTEGER;

PROCEDURE Forces_Of_Vladel
(Arg vladel.vladel_name%TYPE);

PROCEDURE Inform_About_Vladel
(Arg Forces.Force_Name%TYPE);

PROCEDURE Inform_About_Zabegs
(Arg Zabeg.zabeg_data%TYPE);

PROCEDURE Liders_Zabeg;

PROCEDURE New_Zabeg
(Arg1 zabeg.zabeg_id%TYPE,
Arg2 zabeg.zabeg_data%TYPE,
Arg3 zabeg.zabeg_time%TYPE,
Arg4 zabeg.ippodrom_id%TYPE,
```

3.2. Пример выполнения лабораторной работы

```
        Arg5 zabeg.zabeg_name%TYPE,  
        Arg6 results_zabeg.rez_id%TYPE,  
        Arg7 results_zabeg.force_id%TYPE,  
        Arg8 results_zabeg.gokey_id%TYPE,  
        Arg9 results_zabeg.result%TYPE,  
        Arg10 results_zabeg.rez_time%TYPE);  
  
end;
```

Создание тела пакета:

```
CREATE OR REPLACE PACKAGE BODY PACZABEG  
AS  
FUNCTION Kolvo_Forces_Of_Vladel  
    (Arg Forces.vladel_id%TYPE)  
RETURN INTEGER IS  
n INTEGER;  
BEGIN  
    Select COUNT(*) INTO n From Forces Where  
        vladel_id=Arg;  
RETURN n;  
END;  
  
PROCEDURE Forces_Of_Vladel  
    (Arg Vladel.vladel_name%TYPE)  
IS  
T1 Forces.Force_Name%TYPE;  
Cursor CurForce1 IS  
    SELECT Forces.Force_name  
    FROM forces  
    WHERE vladel_id=(select vladel_id from vladel  
                        where vladel_name=arg);  
  
BEGIN  
    Open CurForce1;  
    FETCH CurForce1 INTO T1;  
    WHILE CurForce1%FOUND LOOP  
        DBMS_OUTPUT.PUT_LINE(CurForce1%ROWCOUNT||T1);  
        FETCH CurForce1 INTO T1;  
    END LOOP;  
    Close CurForce1;  
END;  
  
PROCEDURE Inform_About_Vladel
```

```

        (Arg Forces.Force_Name%TYPE)
IS
    T1 Vlade1.vlade1_name%TYPE;
    T2 Vlade1.vlade1_address%TYPE;
    T3 Vlade1.vlade1_telephon%TYPE;
    Cursor CurVlade11 IS
        SELECT vlade1_name,vlade1_address,vlade1_telephon
        FROM Vlade1
        WHERE vlade1_id=(Select vlade1_id
            From Forces WHERE force_name=Arg);

BEGIN
    Open CurVlade11;
    FETCH CurVlade11 INTO T1,T2,T3;
    WHILE CurVlade11%FOUND LOOP
        DBMS_OUTPUT.PUT_LINE (CurVlade11%ROWCOUNT
            || '-Name' || T1 || '-Address ' || T2 || '-Telephon' || T3);
        FETCH CurVlade11 INTO T1,T2,T3;
    END LOOP;
    Close CurVlade11;
END;

PROCEDURE Inform_About_Zabegs
    (Arg Zabeg.zabeg_data%TYPE)
IS
    T1 Forces.force_name%TYPE;
    T2 Gokey.gokey_name%TYPE;
    T3 results_zabeg.result%TYPE;
    T4 results_zabeg.rez_time%TYPE;
    Cursor CurZabeg1 IS
        SELECT Forces.force_name,Gokey.gokey_name,
            results_zabeg.result,results_zabeg.rez_time
        FROM Forces,Gokey,results_zabeg
        WHERE (Forces.force_id=results_zabeg.force_id)AND
            (Gokey.gokey_id=results_zabeg.gokey_id)AND
            results_zabeg.zabeg_id=(select zabeg_id
                from zabeg where zabeg_data=arg);

BEGIN
    Open CurZabeg1;
    FETCH CurZabeg1 INTO T1,T2,T3,T4;
    WHILE CurZabeg1%FOUND LOOP

```

3.2. Пример выполнения лабораторной работы

```
DBMS_OUTPUT.PUT_LINE (CurZabeg1%ROWCOUNT
                        || '-ForceName ' || T1 || '-Gokey '
                        || T2 || '-Mesto' || T3 || '-Time' || T4);
FETCH CurZabeg1 INTO T1,T2,T3,T4;
END LOOP;
Close CurZabeg1;
END;

PROCEDURE Lidere_Zabeg
IS
T1 Forces.Force_name%Type;
T2 Gokey.gokey_name%TYPE;
T3 results_zabeg.rez_time%Type;
T4 zabeg.zabeg_data%Type;
Cursor CurLider1 IS
  Select Forces.Force_name,Gokey.gokey_name,
         results_zabeg.rez_time,zabeg.zabeg_data
  FROM Forces,Gokey,results_zabeg,zabeg
  WHERE (results_zabeg.result=1)AND
         forces.force_id=results_zabeg.force_id)AND
         (gokey.gokey_id=results_zabeg.gokey_id)AND
         (zabeg.zabeg_id=results_zabeg.zabeg_id);
BEGIN
  Open CurLider1;
  FETCH CurLider1 INTO T1,T2,T3,T4;
  WHILE CurLider1%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE (CurLider1%ROWCOUNT
                          || '- ' || T1 || '- ' || T2 || '- ' || T3 || '- ' || T4);
    FETCH CurLider1 INTO T1,T2,T3,T4;
  END LOOP;
  Close CurLider1;
END;

PROCEDURE New_Zabeg
(Arg1 zabeg.zabeg_id%TYPE,
Arg2 zabeg.zabeg_data%TYPE,
Arg3 zabeg.zabeg_time%TYPE,
Arg4 zabeg.ippodrom_id%TYPE,
Arg5 zabeg.zabeg_name%TYPE,
Arg6 results_zabeg.rez_id%TYPE,
Arg7 results_zabeg.force_id%TYPE,
```

```

Arg8 results_zabeg.gokey_id%TYPE,
Arg9 results_zabeg.result%TYPE,
Arg10 results_zabeg.rez_time%TYPE
)
IS
BEGIN
    INSERT INTO zabeg
    VALUES (arg1,arg2,arg3,arg4,arg5);
    COMMIT;
    INSERT INTO results_zabeg
    VALUES (arg6,arg1,arg7,arg8,arg9,arg10);
    COMMIT;
END;
END;
/

```

Создание триггеров:

```

CREATE OR REPLACE TRIGGER TRIG_Forces_Pol
BEFORE UPDATE ON Pol
FOR EACH ROW
BEGIN
    IF (:old.force_pol<>:new.force_pol) THEN
        UPDATE Forces SET force_pol=:new.force_pol
        WHERE force_pol=:old.force_pol;
    END IF;
END;

CREATE OR REPLACE TRIGGER TRIG_Forces_Vladel
BEFORE UPDATE ON vladel
FOR EACH ROW
BEGIN
    IF (:old.vladel_id<>:new.vladel_id) THEN
        UPDATE Forces SET vladel_id=:new.vladel_id
        WHERE vladel_id=:old.vladel_id;
    END IF;
END;

CREATE OR REPLACE TRIGGER TRIG_Zabeg_Ipp
BEFORE UPDATE ON Ippodrom
FOR EACH ROW
BEGIN

```

3.2. Пример выполнения лабораторной работы

```
IF (:old.ippodrom_id<>:new.ippodrom_id) THEN
  UPDATE zabeg SET ippodrom_id=:new.ippodrom_id
  WHERE ippodrom_id=:old.ippodrom_id;
END IF;
END;

CREATE OR REPLACE TRIGGER TRIG_RezZabeg_Force
BEFORE UPDATE ON Forces
FOR EACH ROW
BEGIN
  IF (:old.force_id<>:new.force_id) THEN
    UPDATE results_zabeg SET force_id=:new.force_id
    WHERE force_id=:old.force_id;
  END IF;
END;

CREATE OR REPLACE TRIGGER TRIG_RezZabeg_Gokey
BEFORE UPDATE ON gokey
FOR EACH ROW
BEGIN
  IF (:old.gokey_id<>:new.gokey_id) THEN
    UPDATE results_zabeg SET gokey_id=:new.gokey_id
    WHERE gokey_id=:old.gokey_id;
  END IF;
END;
/
```

3.2.6. Работа с пакетом

Приведем примеры вызова процедур и функции этого пакета.

Для вызова функции **Kolvo_Forces_Of_Vlade1** необходимо установить среду, а затем воспользоваться следующим оператором:

```
SQL> exec DBMS_OUTPUT.PUT_LINE
      (TO_CHAR(PACZABEG.Kolvo_Forces_Of_Vlade(1)));
```

На экране появятся число, характеризующее количество лошадей, имеющих у владельца с номером 1.

Для вызова процедуры **Forces_Of_Vlade1** необходимо воспользоваться следующим оператором:

Примеры вызова процедур и функции пакета

```
SQL> exec PACZABEG. Forces_Of_Vladel ('Petrov');
```

На экране появится список кличек всех лошадей, имеющих у данного владельца с фамилией 'Petrov'.

Для вызова процедуры **Inform_About_Vladel** необходимо воспользоваться следующим оператором:

```
SQL>exec PACZABEG. Inform_About_Vladel('Strela');
```

На экране появится информация о владельце лошади с кличкой 'Strela'.

Для вызова процедуры **Inform_About_Zabegs** необходимо воспользоваться следующим оператором:

```
SQL>exec PACZABEG. Inform_About_Zabegs('1-feb-03');
```

На экране появится информация о забеге в указанную дату '1-feb-03'. Формат даты может быть задан и другим образом, для этого необходимо сделать запрос

```
SQL>select sysdate from dual;
```

Для вызова процедуры **Liders_Zabeg** необходимо воспользоваться следующим оператором:

```
SQL>exec PACZABEG. Liders_Zabeg;
```

В результате на экране появятся лидеры всех заездов.

Для вызова процедуры **New_Zabeg** необходимо воспользоваться следующим оператором:

```
SQL> exec PACZABEG. New_Zabeg(8,'1-nov-03','11:00',  
                             1,'November',11,8,11,1,12.56);
```

В результате будут сделаны добавления в двух таблицах **Забег** и **Результаты забегов**.

СПИСОК ЛИТЕРАТУРЫ

1. Дейт К. Дж. Введение в системы баз данных. — 7-е изд. — М.: Издательский дом “Вильямс”, 2001. — 1072 с.: ил.
2. Конноли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е изд. — М.: Издательский дом “Вильямс”, 2003. — 1440 с.: ил.
3. Астахова И.Ф., Потапов А.С., Чулюков В.А., Журбин А.Н. Информационные системы: Учеб. пособие — Воронеж: Воронежский государственный педагогический университет, 2002. — 148 с.; ил.
4. Четвериков В.Н., Ревунков Г.И., Самохвалов Э.Н. Базы и банки данных: Учеб. для ВУЗов по спец. “АСУ” /Под ред. В.Н. Четверикова. — М.: Высш. шк., 1987. — 248 с.: ил.
5. Rothnie J.B., Goodman N. A Survey of Research and Development in Distributed Database Management //Proc. 3rd Int. Conf. on Very Large Data Bases. — Tokyo, Japan, October, 1977.
6. Пейдж В.Г., Хьюз Н., Остин Д. и др. Использование ORACLE 8 — Киев; М.; СПб.: Издательский дом “Вильямс”, 1998. — 750 с.
7. Смирнов С.Н. Работаем с ORACLE.— М.: Гелиос, 1998.- 318 с.
8. Скотт У. ORACLE 8. Программирование на языке PL/SQL. — М.: Лори, 1997. — 607 с.

Учебное издание

**Астахова Ирина Фёдоровна
Потапов Александр Сергеевич
Чулюков Владимир Алексеевич
Стариков Василий Николаевич**

Практикум по информационным системам. ORACLE

Главный редактор Ю. С. Ковтанюк
Научный редактор И. Ф. Астахова
Литературный редактор Т. Н. Ковтанюк
Ответственный редактор С. В. Соловьян

Подписано в печать 10.04.2004 г. Формат 60 × 84 1/16.
Бумага газетная. Гарнитура Таймс. Печать офсетная.
Усл. печ. л. 10,46. Уч.-изд. л. 6,79.
Тираж 3000 экз. Заказ № 69

Издательство “Юниор”, Украина, 03142, г. Киев, ул. Стуса, 35-37, оф. 111
тел./факс: (044) 452-82-22; e-mail: office@junior.com.ua; <http://www.junior.com.ua>
Свидетельство о внесении субъекта издательского дела в Государственный реестр из-
дателей, производителей и распространителей издательской продукции:
серия ДК, № 368 от 20 марта 2001 г.

Отпечатано с готовых диапозитивов в ООО „Дизайн-студия „Папуга”
79054, г. Львов, ул. Любинская, 92