#### Министерство науки и высшего образования Российской Федерации

#### Федеральное государственное бюджетное образовательное учреждение

#### высшего образования

#### «Владимирский государственный университет

#### имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №2.1

по дисциплине

"Бизнес анализ данных"

Разработка многомерного хранилища. Технологии OLAP

Выполнил:

ст. гр. ПРИм-124

Парахин К.В.

Приняла:

Преподаватель

кафедры ИСПИ

Тимофеев А.А.

Владимир, 2025 г.

Цель работы

Ознакомиться с принципами проектирования OLTP и OLAP систем.

Задание

Необходимо выбрать dataset данных в формате .csv, в котором хотя бы 1 млн записей, определить решаемую задачу аналитической подсистемы и описать предметную область, а также выделить 4 аналитических запроса/сценария использования.

Далее построить UML диаграмму классов OLTP системы, за ней построить схему OLAP куба, затем ER-диаграмму и скрипт реальной БД, попробовать выполнить один из сценариев на тестовом наборе данных

И выполнить сравнение полученных схем данных для OLTP и OLAP систем.

Выполнение работы

1. В качеству dataset была выбрана выборка данных о спортивных событиях букмекерской конторы Betfair – расположенный по ссылке: <https://www.kaggle.com/datasets/zygmunt/betfair-sports>

Он представлен в формате .csv, содержит чуть более 1 млн записей  
В нем представлены следующие колонки: sport\_id, event\_id, settled\_date, full\_description, scheduled\_off, event, dt actual\_off, selection\_id, selection, odds.

1. Данный датасет содержит информацию о ставках, сделанных на спортивные события через букмекерскую контору Betfair. Включает данные о различных видах спорта (sport\_id), событиях (event\_id), времени начала матчей (scheduled\_off, dt actual\_off), описании событий (full\_description, event), исходах (selection) и коэффициентах (odds). Также указаны даты завершения ставок (settled\_date) и идентификаторы исходов (selection\_id).

Решаемая задача в области анализа данных - анализ поведения пользователей и эффективности ставок на основе исторических данных

1. Основные сценарии использования аналитической подсистемы

* За определенный период времени предоставить список самых популярных видов спорта – согласно тому, сколько ставок было сделано на проведенные события этого вида спорта.
* За месяц предоставить пар список идентификаторов спортивных матчей (заданного вида спорта) и типов исходов – в которых на какие то исходы были рассчитаны аномальные коэффициенты – то есть различающиеся от медианного значения коэффициента определенного вида исхода за все время;
* За месяц предоставить набор самых популярных видов исходов, на которые игроки чаще всего ставили свои ставки;
* За месяц найти спортивные матчи, в которых отклонение реального времени начала от запланированного более среднего рассчитанной разницы для матчей конкретного вида спорта – вывести данные в виде группировки идентификаторов по видам спорта, с сортировкой от вида спорта с наибольшим кол-вом таких матчей – к тому, где матчей с таким отклонением наибольшее кол-во, предоставить информацию по первым 10 видам спорта, попавшим в выборку.

1. Далее необходимо промоделировать транзакционную систему данных, на основе вычислений и операций в которой создавался данный dataset

Для этого построим UML Class Diagram с помощью описательного языка PlantUML (содержит сущности Sport, Event, Selection, Bet):

@startuml

class Sport {

+sport\_id: int

+name: string

}

class Event {

+event\_id: int

+sport\_id: int

+full\_description: string

+event: string

+scheduled\_off: datetime

+dt\_actual\_off: datetime

}

class Selection {

+selection\_id: int

+event\_id: int

+selection: string

+odds: float

}

class Bet {

+bet\_id: int

+event\_id: int

+selection\_id: int

+settled\_date: datetime

}

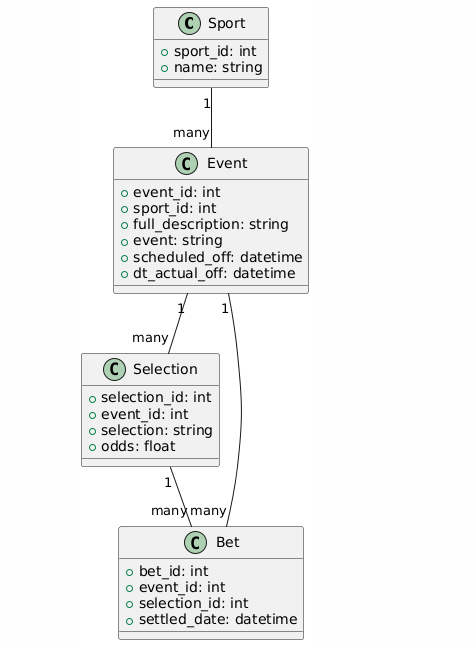
Sport "1" -- "many" Event

Event "1" -- "many" Selection

Event "1" -- "many" Bet

Selection "1" -- "many" Bet

@enduml



1. Далее представим, что мы пытаемся получить схему аналитической системы для этих же данных. Такие данные удобно можно представить в рамках OLAP-куба, который состоит из единственной таблицы фактов: Fact\_Bets. Эта таблица содержит в себе информацию о ставках, сделанных пользователями – и их характеристиках  
     
   Для данной таблицы фактов можно выделить следующие 3 меры:

- total\_bets\_per\_sport (общее количество ставок спорта x) – под этим имеется метрика, связанная с агрегатной функции Count(\*) where sport\_id = x) по таблице bets

- avg\_odds\_per\_selection (средний коэффициент исхода x) – под этим имеется метрика, связанная с агрегатной функцией AVG(odds) where selection\_id = x)

- total\_events\_per\_selection (общее число спортивных событий спорта x) – под этим имеется метрика, связанная с агрегатной функцией Count(\*) where sport\_id = x – по таблице events  
- average\_start\_delay \_per\_sport (среднее отклонение реального начала события от запланированного) – под этим имеется метрика, связанная с агрегатной функцией AVG(actual\_start – scheduled\_start) where sport\_id = x – по таблице events

В качестве измерений можно взять такие:

time\_dimension (временное иерархическое измерение: шкала времени в определенных единицах измерения – уровни связанные с годом, месяцем, днем недели, днем впринципе),

sport\_dimension (измерение: каждое событие относится к определенному виду спорта),

bet\_dimension (тип ставки),

selection\_dimension (вариант исхода ставки)

1. Далее составим физическую схему базы данных – в виде ER – диаграммы на описательном языке Plant UML

@startuml

hide circle

skinparam linetype ortho

' Сущности

entity Sport {

\*sport\_id : integer

--

\*name : varchar

}

entity Event {

\*event\_id : integer

--

\*sport\_id : integer

\*full\_description : text

\*event : varchar

\*scheduled\_off : timestamp

\*dt\_actual\_off : timestamp

}

entity Selection {

\*selection\_id : integer

--

\*event\_id : integer

\*selection : varchar

\*odds : float

}

entity Bet {

\*bet\_id : integer

--

\*event\_id : integer

\*selection\_id : integer

\*settled\_date : timestamp

}

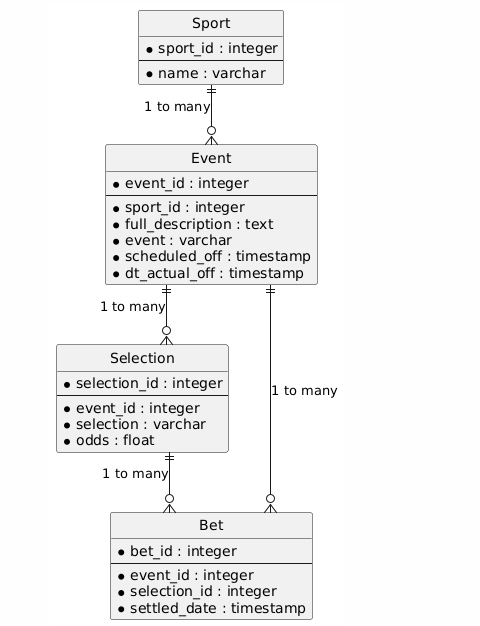
Sport ||--o{ Event : "1 to many"

Event ||--o{ Selection : "1 to many"

Event ||--o{ Bet : "1 to many"

Selection ||--o{ Bet : "1 to many"

@enduml



И на примере реляционной СУБД Postgre SQL представим скрипт БД:

CREATE TABLE Sport (

sport\_id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL

);

CREATE TABLE Event (

event\_id SERIAL PRIMARY KEY,

sport\_id INT REFERENCES Sport(sport\_id),

full\_description TEXT,

event VARCHAR(200),

scheduled\_off TIMESTAMP,

dt\_actual\_off TIMESTAMP

);

CREATE TABLE Selection (

selection\_id SERIAL PRIMARY KEY,

event\_id INT REFERENCES Event(event\_id),

selection VARCHAR(200),

odds FLOAT

);

CREATE TABLE Bet (

bet\_id SERIAL PRIMARY KEY,

event\_id INT REFERENCES Event(event\_id),

selection\_id INT REFERENCES Selection(selection\_id),

settled\_date TIMESTAMP

);

1. Далее я заполнил созданную БД тестовыми данными из 100 первых столбцов исходного dataset) и попробовал использовать предложенный ранее аналитический запрос – а именно вывод топ 5 популярных видов спорта среди введенных данных  
     
   Скрипт:

SELECT

S.name AS sport\_name,

COUNT(B.bet\_id) AS total\_bets

FROM

Bet B

JOIN Event E ON B.event\_id = E.event\_id

JOIN Sport S ON E.sport\_id = S.sport\_id

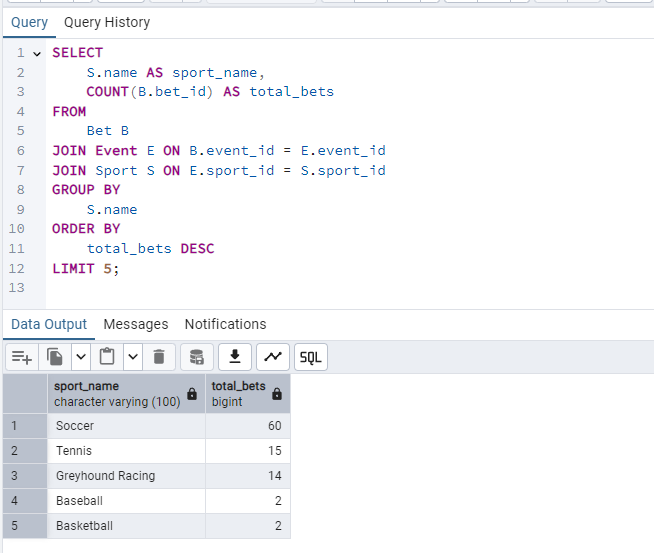
GROUP BY

S.name

ORDER BY

total\_bets DESC

LIMIT 5;



Запрос получения среднего отклонения начала спортивного события от запланированного – в зависимости от вида спорта:

SELECT

s.sport\_id,

s.name,

AVG(ABS(EXTRACT(EPOCH FROM (e.dt\_actual\_off - e.scheduled\_off)))) AS avg\_delay\_seconds

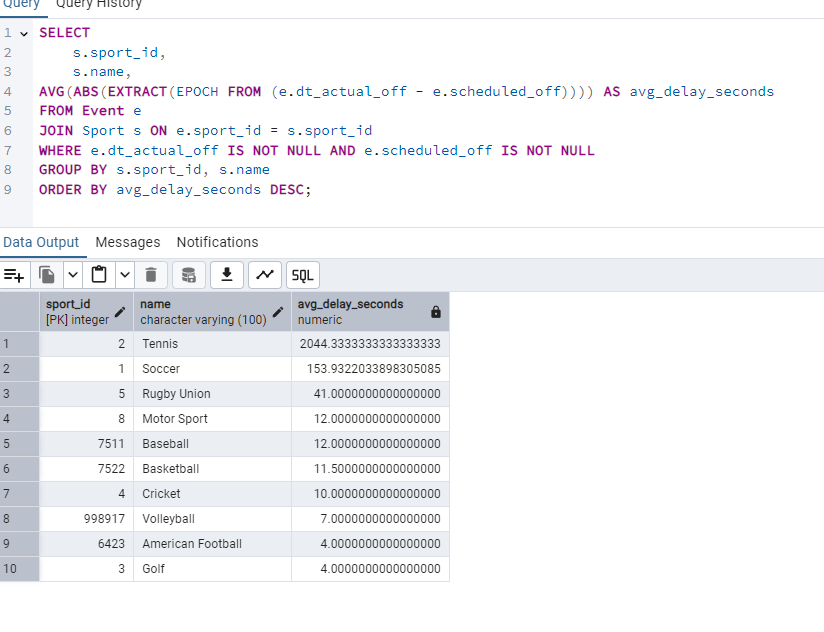
FROM Event e

JOIN Sport s ON e.sport\_id = s.sport\_id

WHERE e.dt\_actual\_off IS NOT NULL AND e.scheduled\_off IS NOT NULL

GROUP BY s.sport\_id, s.name

ORDER BY avg\_delay\_seconds DESC;



1. Рассмотрим проектирование схем данных для OLAP модели, в виде схемы «звезда» и «снежинка»

Для схемы «звезда» создадим центральную таблицу фактов (которая будет хранить измеряемые показатели, ставки и их коэффициенты) и таблицу измерений (доп. характеристики: вид спорта, событие, ставка, дата ставки)

Таблица фактов:

CREATE TABLE Fact\_Bets (

bet\_id SERIAL PRIMARY KEY,

event\_id INT,

selection\_id INT,

sport\_id INT,

date\_id INT,

odds FLOAT,

settled\_date TIMESTAMP,

bet\_amount DECIMAL(10,2),

payout DECIMAL(10,2));

Таблицы измерений:

CREATE TABLE Dim\_Sport (

sport\_id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL);

CREATE TABLE Dim\_Event (

event\_id SERIAL PRIMARY KEY,

sport\_id INT REFERENCES Dim\_Sport(sport\_id),

event VARCHAR(200),

scheduled\_off TIMESTAMP,

dt\_actual\_off TIMESTAMP);

CREATE TABLE Dim\_Selection (

selection\_id SERIAL PRIMARY KEY,

event\_id INT REFERENCES Dim\_Event(event\_id),

selection VARCHAR(200));

CREATE TABLE Dim\_Date (

date\_id SERIAL PRIMARY KEY,

full\_date DATE,

year INT,

month INT,

day INT,

weekday VARCHAR(10));

Благодаря такой своеобразной структуре данная БД может помочь аналитике в более сложном аналитическом запросе, например, получение самых популярных видов ставок по видам спорта за промежуток времени:

SELECT

s.name AS sport\_name,

sel.selection,

dat.full\_date,

ev.event

COUNT(\*) AS bet\_count

FROM Fact\_Bets fb

JOIN Dim\_Sport s ON fb.sport\_id = s.sport\_id

JOIN Dim\_Selection sel ON fb.selection\_id = sel.selection\_id

JOIN Dim\_Event ev ON fb.event\_id = ev.event\_id  
 JOIN Dim\_date dat ON fb.date\_id = dat.date\_id  
 WHERE dat.full\_date < NOW() - 'interval::30 days'

GROUP BY s.name, sel.selection

ORDER BY bet\_count DESC

LIMIT 10;

Для схемы «снежинка» оставим ту же таблицу фактов, но дополнительно проведем нормализацию таблиц измерений – а именно выделим отдельные таблицы: Dim\_Event\_Type, Dim\_Selection\_Type, Dim\_Sport\_Category, а также уберем избыточные данные из таблицы Events – для оптимизации хранения данных

Измененные таблицы измерений:

CREATE TABLE Dim\_Sport (

sport\_id SERIAL PRIMARY KEY,

category\_id INT,

name VARCHAR(100) NOT NULL);

CREATE TABLE Dim\_Sport\_Category (

category\_id SERIAL PRIMARY KEY,

category\_name VARCHAR(100));

CREATE TABLE Dim\_Event (

event\_id SERIAL PRIMARY KEY,

sport\_id INT REFERENCES Dim\_Sport(sport\_id),

event\_type\_id INT,

scheduled\_off TIMESTAMP,

dt\_actual\_off TIMESTAMP);

CREATE TABLE Dim\_Event\_Type (

event\_type\_id SERIAL PRIMARY KEY,

type\_name VARCHAR(100));

CREATE TABLE Dim\_Selection (

selection\_id SERIAL PRIMARY KEY,

event\_id INT REFERENCES Dim\_Event(event\_id),

selection\_type\_id INT);

CREATE TABLE Dim\_Selection\_Type (

selection\_type\_id SERIAL PRIMARY KEY,

selection VARCHAR(200));

Благодаря такой своеобразной структуре данная БД может помочь аналитике в еще более сложном аналитическом запросе, например, получение выборки аномальных коэффициентов на исходы спортивных событий:

SELECT

sc.category\_name AS sport\_category,

s.name AS sport\_name,

et.type\_name AS event\_type,

sel\_type.selection AS selection\_type,

AVG(fb.odds) AS avg\_odds,

PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY fb.odds) AS median\_odds,

(AVG(fb.odds) - PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY fb.odds)) AS deviation

FROM Fact\_Bets fb

JOIN Dim\_Sport s ON fb.sport\_id = s.sport\_id

JOIN Dim\_Sport\_Category sc ON s.category\_id = sc.category\_id

JOIN Dim\_Event e ON fb.event\_id = e.event\_id

JOIN Dim\_Event\_Type et ON e.event\_type\_id = et.event\_type\_id

JOIN Dim\_Selection sel ON fb.selection\_id = sel.selection\_id

JOIN Dim\_Selection\_Type sel\_type ON sel.selection\_type\_id = sel\_type.selection\_type\_id

GROUP BY sc.category\_name, s.name, et.type\_name, sel\_type.selection

HAVING ABS(AVG(fb.odds) - PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY fb.odds)) > 0.5

ORDER BY deviation DESC;

1. По итогу, получилось 2 схемы данных о ставках, выполненные в рамках OLTP и OLAP подхода

Каждая из схем имеет свои преимущества/недостатки – и вполне может быть использована в конкретных ситуациях

OLTP схема имеет преимущества: минимизирует избыточность данных (нормализует данные), предоставляет возможность проведения транзакционных операций записи/обновления/удаления записей из БД, благодаря использования внешних ключей обеспечивает ссылочную целостность данных. Но в недостатках есть следующее: достаточно сложно использовать для комплексных аналитических запросов (типа запросов на события с аномальными коэффициентами) – удобно для несложных запросов, типа самых популярных видов спортов.

OLAP схема предоставляет богатый инструмент для аналитических исследований, может содержать и оперировать большими массивами данных, поддерживает многомерные анализы и сложные сценарии. Но ее сложно модифицировать, особенно что то обновлять или удалять, она не работает в параллельных сценариях, во многом дублирует некоторые данные и создает избыточность.

Вывод

В результате выполнения работы были изучены принципы проектирования OLTP и OLAP систем.