#### 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).

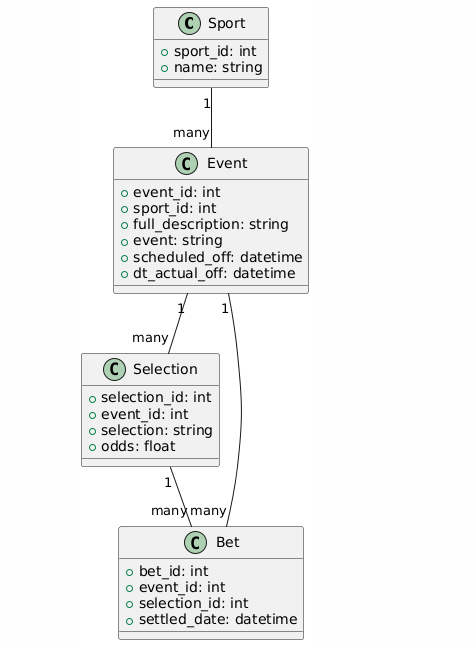
Решаемая задача в области анализа данных - анализ поведения пользователей и эффективности ставок на основе исторических данных

1. Основные сценарии использования аналитической подсистемы

* Предоставить список самых популярных видов спорта – согласно тому, сколько ставок было сделано на проведенные события этого вида спорта. Представить в разрезе вид спорта + тип спорта
* За год предоставить список сделанных ставок для выбранного спорта – у которых отклонение коэффициента больше среднего коэффициента для типа исхода, на который делается ставка, отсортировать ставки в порядке уменьшения отклонения коэффициента.
* За год предоставить набор самых популярных видов исходов, на которые игроки чаще всего ставили свои ставки - отсортировать и представить выборку из 10 исходов с самым большим количеством выполненных ставок, в порядке уменьшения количества
* За год найти спортивные матчи заданного вида спорта, в которых отклонение реального времени начала от запланированного более среднего рассчитанной разницы для матчей конкретного вида спорта – вывести данные в виде сортированного списка матчей – с указанием расхождения значения отклонения от среднего.

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):



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:

- events\_count (общее количество матчей спорта x) – под этим имеется метрика, связанная с агрегатной функции Count(\*) where sport\_id = x) по таблице sports

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

- total\_plays\_count (общее число поставленных ставок определенного исхода) – под этим имеется метрика, связанная с агрегатной функцией Count(\*) where selection\_id = x в таблице bets

Так как аналитически востребовано получать информацию по этой мере в разрезе года – то необходимо добавить еще total\_plays\_count\_per\_year  
- average\_start\_diff (среднее отклонение реального начала события от запланированного для спорта x) – под этим имеется метрика, связанная с агрегатной функцией AVG(actual\_start – scheduled\_start) where sport\_id = x – по таблице events. Таблица events при этом имеет поле actual\_start\_diff – которое уже содержит вычисленную разницу между реальным и назначенным временем начала матча.

В качестве измерений можно взять такие:

dim\_date (временное иерархическое измерение: шкала времени в определенных единицах измерения – содержит иерархическое измерение, связанное с годом),

dim\_sport (измерение: каждое событие относится к определенному виду спорта и типу спорта – иерархическое измерение к dim\_sport\_type),

dim\_selection (вариант исхода ставки)

Также нужно добавить еще измерение для исходов в разрезе временного промежутка (года) – dim\_agg\_selection\_plays\_count\_per\_year (чтобы можно было для каждого исхода selection\_id видеть, сколько было поставлено ставок пользователями в заданный год)

Представим схему OLAP-куба:

CREATE TABLE dim\_sport\_type (

    id BIGINT PRIMARY KEY,

    type\_name VARCHAR(100));

CREATE TABLE dim\_sport (

    id BIGINT PRIMARY KEY,

    name VARCHAR(100) NOT NULL,

    sport\_type\_id BIGINT REFERENCES dim\_sport\_type(id),

    events\_count BIGINT,

    avg\_start\_diff FLOAT);

CREATE TABLE dim\_event (

    id BIGINT PRIMARY KEY,

    sport\_id BIGINT REFERENCES dim\_sport(id),

    event\_name VARCHAR(200),

    actual\_start\_diff interval);

CREATE TABLE dim\_selection (

    id BIGINT PRIMARY KEY,

    selection\_name VARCHAR(40),

    event\_id BIGINT REFERENCES dim\_event(id),

    plays\_count BIGINT,

    median\_odds FLOAT);

CREATE TABLE dim\_agg\_selection\_plays\_count\_per\_year (

    selection\_id BIGINT REFERENCES dim\_selection(id),

    year INT,

    total\_plays\_count BIGINT,

    PRIMARY KEY (selection\_id, year));

CREATE TABLE dim\_date (

    id BIGINT PRIMARY KEY,

    full\_date DATE,

    year INT);

CREATE TABLE fact\_bets (

    id BIGINT PRIMARY KEY,

    event\_id BIGINT REFERENCES dim\_event(id),

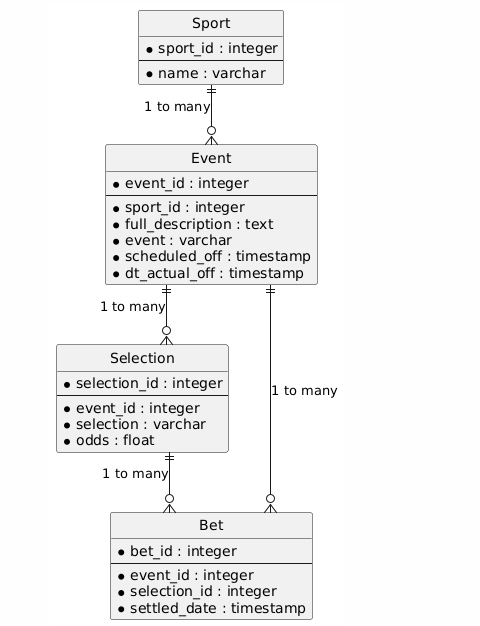
    selection\_id BIGINT REFERENCES dim\_selection(id),

    sport\_id BIGINT REFERENCES dim\_sport(id),

    date\_id BIGINT REFERENCES dim\_date(id),

    odds FLOAT);

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



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 выполнить 4 разных аналитических запроса с БД

Предоставить список самых популярных видов спорта – согласно тому, сколько ставок было сделано на проведенные события этого вида спорта. Представить в разрезе вид спорта + тип спорта

WITH ranked\_sports AS (

    SELECT

        dst.type\_name AS sport\_type,

        ds.name AS sport\_name,

        ds.events\_count,

        DENSE\_RANK() OVER (ORDER BY ds.events\_count DESC) AS overall\_rank

    FROM

        dim\_sport ds

    JOIN

        dim\_sport\_type dst ON ds.sport\_type\_id = dst.id

    WHERE

        ds.events\_count > 0

)

SELECT

    sport\_type,

    sport\_name,

    events\_count,

    overall\_rank AS rank

FROM

    ranked\_sports

WHERE

    overall\_rank <= 10

ORDER BY

    events\_count DESC,

    sport\_type,

    sport\_name;

* За 2014 год предоставить список сделанных ставок для выбранного спорта (теннис) – у которых отклонение коэффициента больше среднего коэффициента для типа исхода, на который делается ставка, отсортировать ставки в порядке уменьшения отклонения коэффициента.

SELECT

    fb.id AS bet\_id,

    ds.median\_odds AS selection\_median\_odds,

    fb.odds AS bet\_odds,

    fb.odds - ds.median\_odds AS odds\_difference

FROM

    fact\_bets fb

JOIN dim\_selection ds ON fb.selection\_id = ds.id

JOIN dim\_date dd ON fb.date\_id = dd.id

JOIN dim\_event de ON fb.event\_id = de.id

JOIN dim\_sport dsp ON de.sport\_id = dsp.id

WHERE

    dsp.name = 'Soccer'

    AND dd.year = 2014

    AND fb.odds > ds.median\_odds

ORDER BY

    odds\_difference DESC;

* За 2014 год предоставить набор самых популярных видов исходов, на которые игроки чаще всего ставили свои ставки в футболе – отсортировать и представить выборку из 10 исходов с самым большим количеством выполненных ставок, в порядке уменьшения количества

WITH soccer\_most\_played\_selections AS (

    SELECT

        dsel.id as selection\_id,

        dsel.selection\_name,

        daspcpy.total\_plays\_count AS bets\_count,

        RANK() OVER (ORDER BY daspcpy.total\_plays\_count DESC) AS popularity\_rank

    FROM

        dim\_agg\_selection\_plays\_count\_per\_year daspcpy

    JOIN dim\_selection dsel ON daspcpy.selection\_id = dsel.id

    JOIN dim\_event de ON dsel.event\_id = de.id

    JOIN dim\_sport ds ON de.sport\_id = ds.id

    WHERE

        ds.name = 'Soccer'

        AND daspcpy.year = 2014

)

SELECT

    selection\_id,

    selection\_name,

    bets\_count

FROM

    soccer\_most\_played\_selections

WHERE

    popularity\_rank <= 10

ORDER BY

    bets\_count DESC;

* За 2014 год найти спортивные теннисные матчи, в которых отклонение реального времени начала от запланированного более среднего рассчитанной разницы для матчей конкретного вида спорта – вывести данные в виде сортированного списка матчей – с указанием расхождения значения отклонения от среднего.

SELECT

    de.id as match\_id,

    de.event\_name AS match\_name,

    de.actual\_start\_diff AS actual\_deviation,

    EXTRACT(EPOCH FROM de.actual\_start\_diff) AS deviation\_seconds,

    EXTRACT(EPOCH FROM ds.avg\_start\_diff) AS avg\_deviation\_seconds,

    EXTRACT(EPOCH FROM (de.actual\_start\_diff - ds.avg\_start\_diff)) AS above\_avg\_seconds

FROM

    dim\_event de

JOIN

    dim\_sport ds ON de.sport\_id = ds.id

WHERE

    ds.name = 'Tennis'

    AND de.actual\_start\_diff IS NOT NULL

    AND de.actual\_start\_diff > ds.avg\_start\_diff

ORDER BY

    above\_avg\_seconds 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.

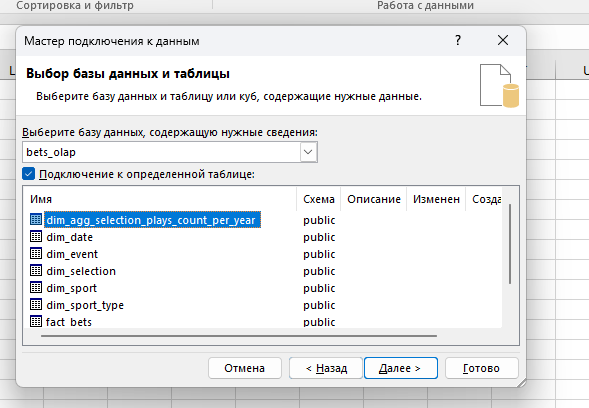
OLAP схема предоставляет богатый инструмент для аналитических исследований, может содержать и оперировать большими массивами данных, поддерживает многомерные анализы и сложные сценарии. Но ее сложно модифицировать, особенно что то обновлять или удалять, она не работает в параллельных сценариях, во многом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

Далее проведем вставку экспериментальных данных о первой сотне ставок из исходного dataset для описанной ранее OLAP модели данных – и попробуем выполнить описанные выше 4 аналитических запроса – а далее провести их визуализацию в BI средстве

В качестве BI средства буду использовать Microsoft Excel + Pivot

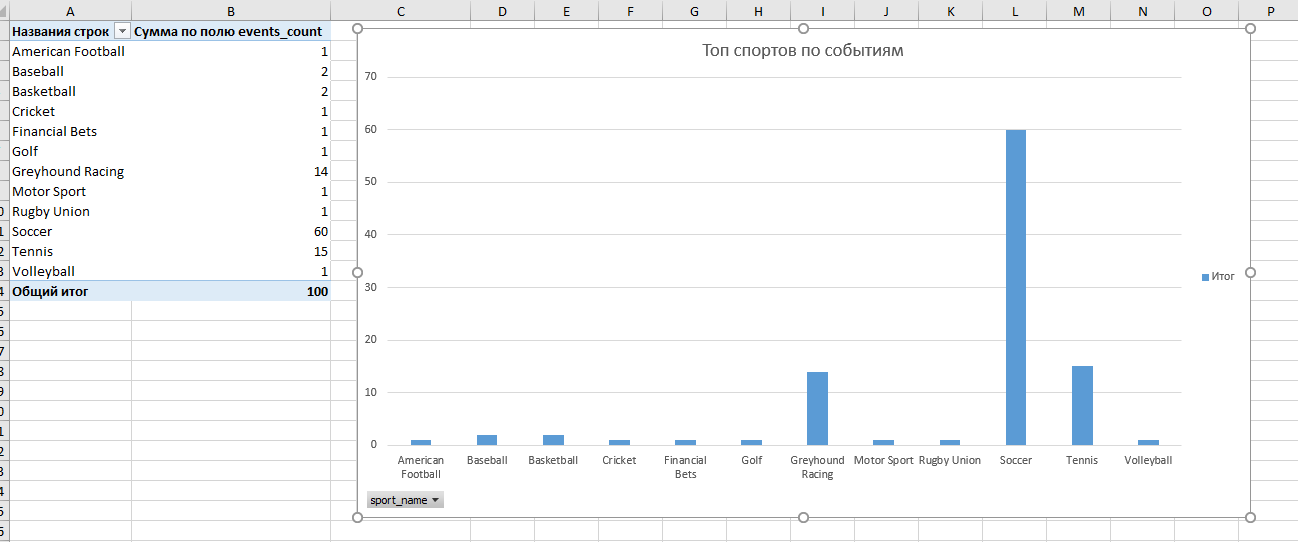
Для начала создаю новый источник PostgreSQL ODBC, подключаю туда свою БД bets\_olap

Используя мастер подключений открываю и вижу таблицы OLAP куба

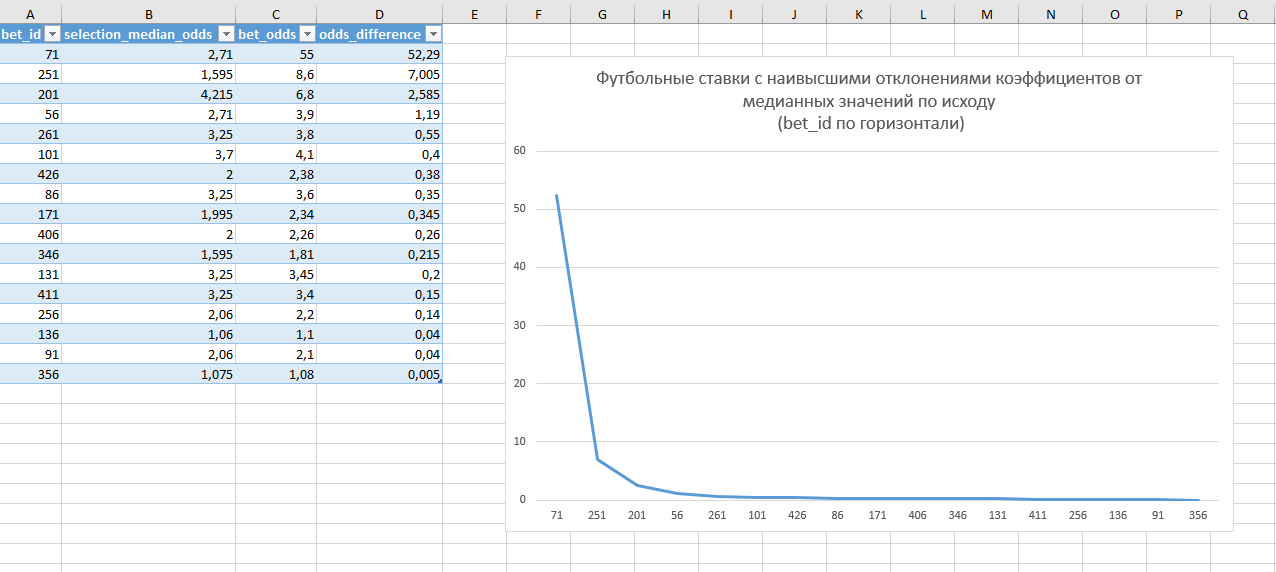


Далее на основе каждого из 4 запросов для удобства импорта полученных данных из выборки в результате запроса – создал внутри БД материализованные представления

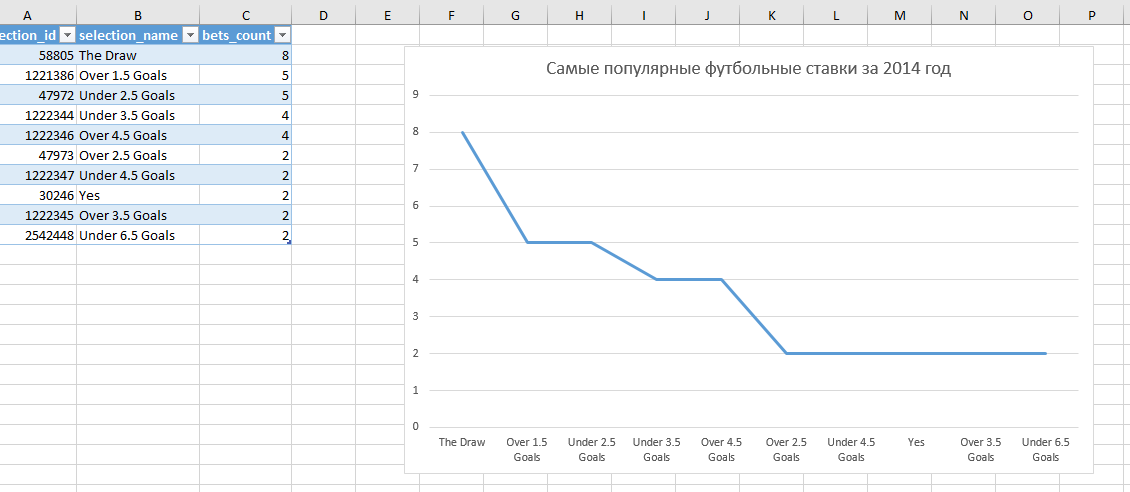
* Предоставить список самых популярных видов спорта – согласно тому, сколько ставок было сделано на проведенные события этого вида спорта.



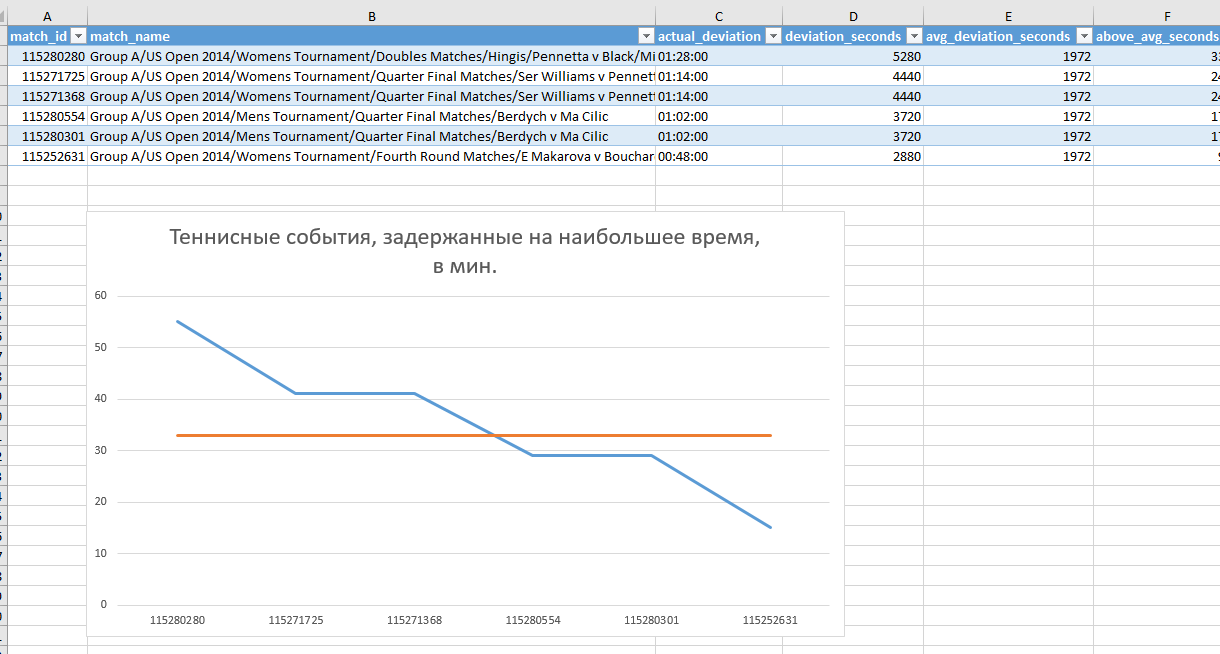
* За 2014 год предоставить список сделанных ставок для выбранного спорта (теннис) – у которых отклонение коэффициента больше среднего коэффициента для типа исхода, на который делается ставка, отсортировать ставки в порядке уменьшения отклонения коэффициента.



* За 2014 год предоставить набор самых популярных видов исходов, на которые игроки чаще всего ставили свои ставки в футболе – отсортировать и представить выборку из 10 исходов с самым большим количеством выполненных ставок, в порядке уменьшения количества



* За 2014 год найти спортивные теннисные матчи, в которых отклонение реального времени начала от запланированного более среднего рассчитанной разницы для матчей конкретного вида спорта – вывести данные в виде сортированного списка матчей – с указанием расхождения значения отклонения от среднего.



Вывод

В результате выполнения работы были изучены принципы проектирования OLTP и OLAP систем на примере работы с данными предметной области, связанной с хранением и анализом ставок на спортивные события.