Semester Project DEMO

Team Name: KeiData

- Tarun Reddi
- Charvi Kusuma

University at Buffalo The State University of New York

Gear Shift Genius: Master of Formula 1 Data Management

University at Buffalo The State University of New York

Agenda

- Introduction to Project
- Database Design and Schema
- Tasks 1-4 Implemented
- Tasks 5-7 Implemented
- Data Management Application Demo

Introduction

Problem Statement

Problem Statement: Enhance efficiency, collaboration, and decision-making across Formula 1 teams and stakeholders.

Current Challenges: Reliance on manual processes and tools like Excel, leading to inefficiencies with speed and limited insights.

Why Transition to a Database?

- Data Integrity
- Real-Time Updates
- Simultaneous Access
- Advanced Analytics

Objectives

- Provide a Centralized System
- Expandable User Interface
- Data Validation and Integrity
- Include Advanced Analytics and Reporting
- Effective Collaboration and Access

Efficiency and Competitiveness: Streamlines workflows, reduces errors, and maximizes data potential.

Innovation and Strategies: Helps teams gain deeper insights, identify improvement areas, and develop competitive strategies.

Accessibility and Continuous Improvement: Makes data access democratic within teams, fostering a culture of continuous improvement.

Target Users

Database Administration

- Managed by team DBAs ensuring data security and integrity.
- Larger teams may have data analysts/scientists for deeper insights.
- FIA manages central database for regulatory and dissemination purposes.

Formula 1 Teams

- Sponsors
- Broadcasters
- Regulatory Bodies

Stakeholders

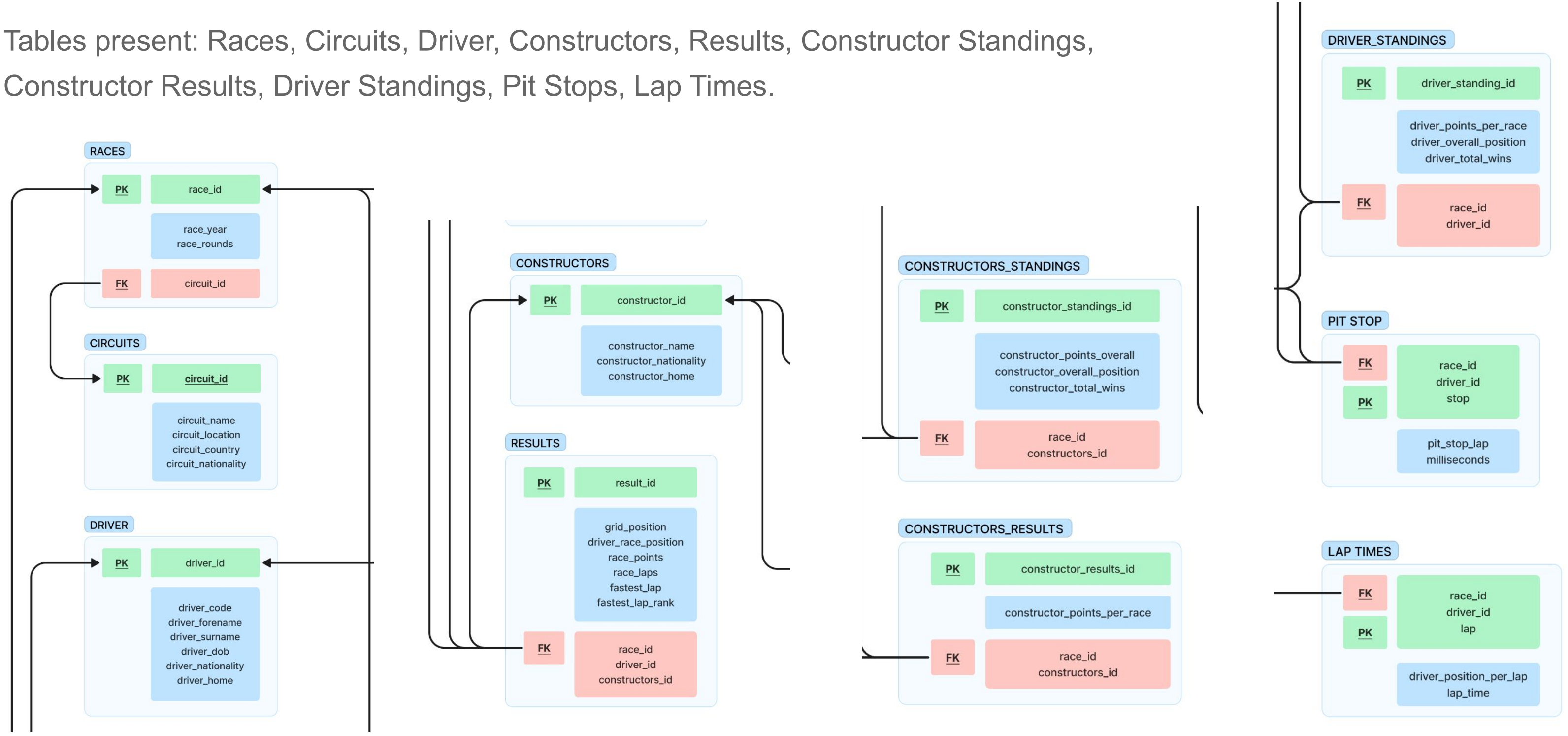
- Engineers
- Strategists
- Data Analysts
- Team Managers

Database Design and Schema

Database Design and Schemas

Source: Formula 1 Dataset[1950 - 2023]

Tables present: Races, Circuits, Driver, Constructors, Results, Constructor Standings, Constructor Results, Driver Standings, Pit Stops, Lap Times.



RACES
PK race_id
race_year
race_rounds
FK circuit_id

CIRCUITS
PK circuit_id
circuit_name
circuit_location
circuit_country
circuit_nationality

DRIVER
PK driver_id
driver_code
driver_forename
driver_surname
driver_dob
driver_nationality
driver_home

CONSTRUCTORS
PK constructor_id
constructor_name
constructor_nationality
constructor_home

RESULTS
PK result_id
grid_position
driver_race_position
race_points
race_laps
fastest_lap
fastest_lap_rank
FK race_id
driver_id
constructors_id

CONSTRUCTORS_STANDINGS
PK constructor_standings_id
constructor_points_overall
constructor_overall_position
constructor_total_wins
FK race_id
constructors_id

CONSTRUCTORS_RESULTS
PK constructor_results_id
constructor_points_per_race
FK race_id
constructors_id

DRIVER_STANDINGS
PK driver_standing_id
driver_points_per_race
driver_overall_position
driver_total_wins
FK race_id
driver_id

PIT STOP
FK
PK
race_id
driver_id
stop
pit_stop_lap
milliseconds

LAP TIMES
FK
PK
race_id
driver_id
lap
driver_position_per_lap
lap_time

Key Points

Relationships

Circuits and Races: One circuit hosts multiple races (1:N).

Races: Belongs to one circuit (N:1). Connected to results, standings, lap times, pit stops, and constructors' results (1:N).

Drivers and Constructors: Both have multiple results and standings entries (1:N). Drivers have multiple lap times and pit stops (1:N).

Results: Each result linked to one race, one driver, and one constructor (N:1).

Standings and Results: Each entry linked to a specific race and either a driver or constructor (N:1).

Participation

Weak Entity: Pit Stop (depends on Races and Drivers).

Total Participation: All main entities are fully involved in relationships via foreign keys.

Tasks Implemented

Phase 1 [Task 1-4]

Phase 2 [Task 5-7]

Tasks 1-4

Task 1

- Selection of Dataset, Identifying use case, kind of queries will be used, updation of data.
- Formula 1 Dataset, Race Strategy Optimization, Performance Analysis, Car Development, Driver Evaluation etc.

Task 2

- Created ER and identified constraints, further converted it to relational schema and applied database design theory.
- Showed the constraints and justified the purpose of Foreign Key or Primary Key.

Task 3

- Database created and inserted.
- DEMO - All tables present in Formula 1 Database. Showing all Tables
- DEMO - We also verified with 4 advanced queries. Showing 1

Task 4

- Listed all the dependencies and provided clear justification of BCNF.

Tasks 5-7

Task 5

- Problems Faced: Slow Query Execution, Sequential Scans
- Solutions Implemented: EXPLAIN command to understand the cost associated
- Created indexes on CircuitID in both the Circuits and Races tables, and on RaceID in the Results table
- Optimization with Subqueries

Task 6

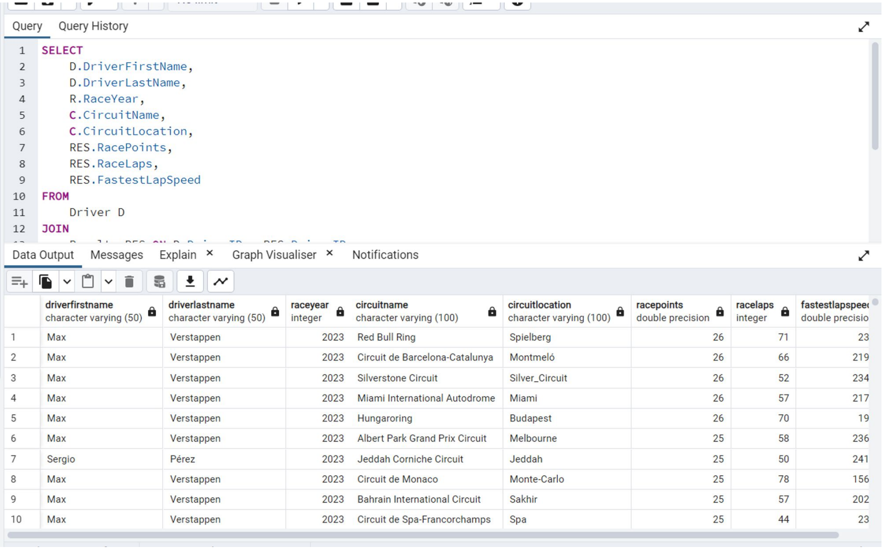
- We worked on 10 different SQL queries.
- DEMO: 1 Insert, 1 Update, 1 Delete, 1 Select

Task 7

- Query execution analysis: 3 Problematic queries, Analysed costs, Optimized with subqueries and indexing.

Task 7 - Problematic Queries

```
SELECT
   D.DriverFirstName, D.DriverLastName,  R.RaceYear, C.CircuitName,
   C.CircuitLocation,  RES.RacePoints, RES.RaceLaps,  RES.FastestLapSpeed
FROM
   Driver D
JOIN   Results RES ON D.DriverID = RES.DriverID
JOIN   Races R ON RES.RaceID = R.RaceID
JOIN   Circuits C ON R.CircuitID = C.CircuitID
WHERE   R.RaceYear = 2023
ORDER BY  RES.RacePoints DESC;
```



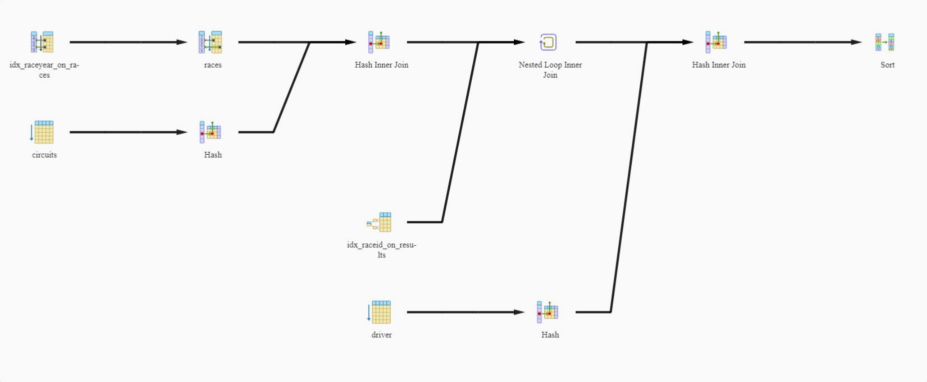
```
SELECT
    D.DriverFirstName,
    D.DriverLastName,
    R.RaceYear,
    C.CircuitName,
    C.CircuitLocation,
    RES.RacePoints,
    RES.RaceLaps,
    RES.FastestLapSpeed
FROM
    Driver D
JOIN
```

	driverfirstname character varying (50)	driverlastname character varying (50)	raceyear integer	circuitname character varying (100)	circuitlocation character varying (100)	racepoints double precision	racelaps integer	fastestlapspeed double precisio
1	Max	Verstappen	2023	Red Bull Ring	Spielberg	26	71	23
2	Max	Verstappen	2023	Circuit de Barcelona-Catalunya	Montmeló	26	66	219
3	Max	Verstappen	2023	Silverstone Circuit	Silver_Circuit	26	52	234
4	Max	Verstappen	2023	Miami International Autodrome	Miami	26	57	217
5	Max	Verstappen	2023	Hungaroring	Budapest	26	70	19
6	Max	Verstappen	2023	Albert Park Grand Prix Circuit	Melbourne	25	58	236
7	Sergio	Pérez	2023	Jeddah Corniche Circuit	Jeddah	25	50	241
8	Max	Verstappen	2023	Circuit de Monaco	Monte-Carlo	25	78	156
9	Max	Verstappen	2023	Bahrain International Circuit	Sakhir	25	57	202
10	Max	Verstappen	2023	Circuit de Spa-Francorchamps	Spa	25	44	23

Task 7 - Problematic Queries

```
CREATE INDEX IF NOT EXISTS idx_driverid_on_driver
        ON Driver (DriverID);
CREATE INDEX IF NOT EXISTS idx_circuitid_on_circuits
        ON Circuits (CircuitID);
ANALYZE Driver, Results, Races, Circuits; VACUUM Driver, Results, Races, Circuits;
```

Reduction of Cost after Indexing: From 315 to 306

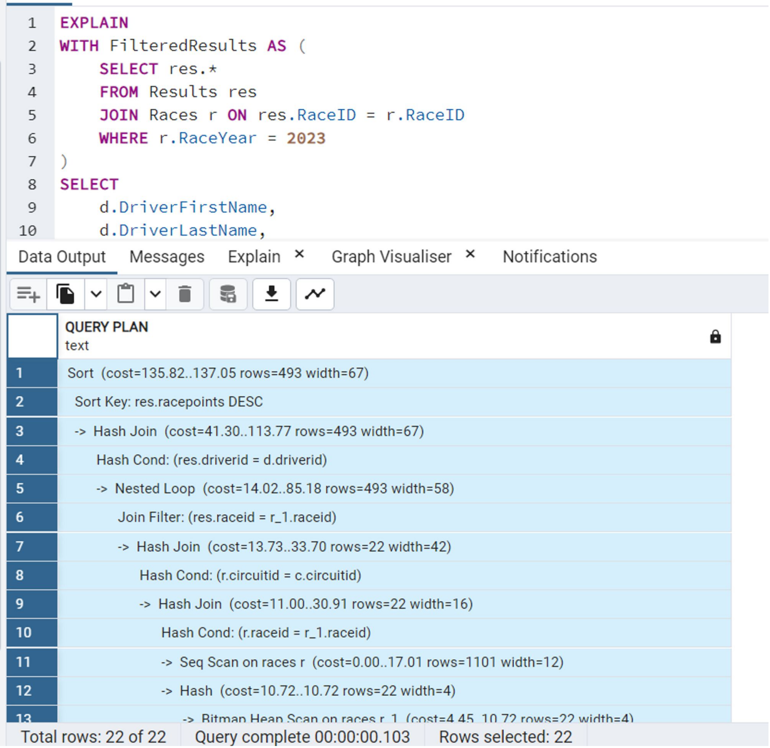


Sequential Joins

Task 7 - Problematic Queries

```
WITH FilteredResults AS ( SELECT res.* FROM Results res
                          JOIN Races r ON res.RaceID = r.RaceID
                          WHERE r.RaceYear = 2023)
SELECT
  d.DriverFirstName, d.DriverLastName, r.RaceYear, c.CircuitName,
  c.CircuitLocation, res.RacePoints, res.RaceLaps, res.FastestLapSpeed
FROM  Driver d
JOIN  FilteredResults res ON d.DriverID = res.DriverID
JOIN  Races r ON res.RaceID = r.RaceID
JOIN  Circuits c ON r.CircuitID = c.CircuitID
ORDER BY  res.RacePoints DESC;
```

Reduction of Cost after pre-filtering the data in subqueries: From 306 to 135



```
EXPLAIN
WITH FilteredResults AS (
    SELECT res.*
    FROM Results res
    JOIN Races r ON res.RaceID = r.RaceID
    WHERE r.RaceYear = 2023
)
SELECT
    d.DriverFirstName,
    d.DriverLastName,
```

```
    QUERY PLAN
    text
1   Sort  (cost=135.82..137.05 rows=493 width=67)
2     Sort Key: res.racepoints DESC
3     ->  Hash Join  (cost=41.30..113.77 rows=493 width=67)
4           Hash Cond: (res.driverid = d.driverid)
5           ->  Nested Loop  (cost=14.02..85.18 rows=493 width=58)
6                 Join Filter: (res.raceid = r_1.raceid)
7                 ->  Hash Join  (cost=13.73..33.70 rows=22 width=42)
8                       Hash Cond: (r.circuitid = c.circuitid)
9                       ->  Hash Join  (cost=11.00..30.91 rows=22 width=16)
10                            Hash Cond: (r.raceid = r_1.raceid)
11                            ->  Seq Scan on races r  (cost=0.00..17.01 rows=1101 width=12)
12                            ->  Hash  (cost=10.72..10.72 rows=22 width=4)
13                                  ->  Bitmap Heap Scan on races r_1  (cost=4.45..10.72 rows=22 width=4)
Total rows: 22 of 22    Query complete 00:00:00.103    Rows selected: 22
```

Data Management Application Demo

University at Buffalo The State University of New York

Thankyou

Team:

- Tarun Reddi
- Charvi Kusuma