University of British Columbia, Vancouver
Department of Computer Science

Project Description

This project is a motorsport database management system that allows efficient tracking and management of the various aspects of motorsport teams, specifically their vehicles, drivers, sponsors, and engineering team. It also allows the tracking of various relationships like a team's participation in racing series, sponsorship amounts and their tier levels, and the assignment of engineers to each team, enabling users to seamlessly add, update, and manage large amounts of data involved in motorsports. For example, the user can track sponsorships and their amounts, records of competition, and engineers for teams. The system also supports four queries that return additional, complex aggregated information. The backend is built using Node.js and Oracle DB, while the frontend is implemented using Next.js. Rigorous testing was carried out through Postman to ensure correctness and relationships between entities were enforced using key constraints.

Final Schema Changes

Several aspects of the schema were revised during development, resulting in a final schema that differs from the one originally turned in.

1. Attributes that were originally assigned the data type VARCHAR and INT, were changed to VARCHAR2 and NUMBER, respectively, as VARCHAR and INT are not supported by the Oracle Database. These are equivalent data types that are compatible with the Oracle Database environment.

2. Check constraints were added to various attributes to enforce valid entries
    a. year_first_produced BETWEEN 1000 AND 2015
    b. engine_cc BETWEEN 100 And 3000
    c. drivetrain IN ('FWD', 'RWD', '4WD', 'AWD')

3. The *name* attribute in **Driver** entity was renamed to *driver_name* to improve clarity and readability

University of British Columbia, Vancouver
Department of Computer Science

Schema & Example Data

Team_Principal

```
CREATE TABLE Team_Principal (
    team_principal VARCHAR2(100) PRIMARY KEY,
    team_name VARCHAR(100) UNIQUE
);
```

team_principal	team_name
Toto Wolf	Mercedes AMG Petronas
Christian Horner	Red Bull Racing
Frederic Vasseur	Scuderia Ferrari
Mike Krack	Aston Martin F1

Team

```
CREATE TABLE Team(
    team_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    team_principal VARCHAR2(100),
    year_founded NUMBER(4) CHECK (year_founded BETWEEN 1000 AND 9999),
    CONSTRAINT fk_team_principal FOREIGN KEY (team_principal)
        REFERENCES Team_Principal (team_principal)
        ON DELETE SET NULL
);
```

team_id	team_principal	year_founded
1	Toto Wolf	1970
2	Christian Horner	2004
3	Frederic Vasseur	1929
4	Mike Krack	2021

University of British Columbia, Vancouver
Department of Computer Science

Sponsor_Tier

```
CREATE TABLE Sponsor_Tier (
    tier_level VARCHAR2(50),
    amount_contributed NUMBER CHECK (amount_contributed >= 0),
    CONSTRAINT sponsor_tier_pk PRIMARY KEY (tier_level),
    CONSTRAINT unique_amount_contributed UNIQUE (amount_contributed)
);
```

tier_level	amount_contributed
Bronze	100000
Silver	500000
Gold	1000000
Platinum	5000000
Diamond	10000000

Sponsor

```
CREATE TABLE Sponsor(
    sponsor_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    sponsor_name VARCHAR2(50) NOT NULL,
    tier_level VARCHAR2(50),
    point_of_contact VARCHAR2(100),
    CONSTRAINT unique_sponsor_name UNIQUE (sponsor_name),
    CONSTRAINT fk_sponsor_tier FOREIGN KEY (tier_level)
        REFERENCES Sponsor_Tier(tier_level)
        ON DELETE CASCADE
);
```

sponsor_id	sponsor_name	tier_level	point_of_contact
1	Petronas	Diamond	Razlan Razali
2	Ineos	Platinum	Jim Ratcliffe
3	Oracle	Diamond	Larry Ellison
4	Tag Heuer	Gold	Frederic Arnault
5	Shell	Platinum	Ben van Beurden
6	Santander	Gold	Ana Botin
7	Aramco	Diamond	Amin H. Nasser
8	Congizant	Platinum	Brian Humphries

Driver

```
CREATE TABLE Driver (
    driver_name VARCHAR2(100) PRIMARY KEY,
    driver_number NUMBER NOT NULL
);
```

driver_name	driver_number
Lewis Hamilton	44
George Russell	63
Max Verstappen	1
Sergio Perez	11
Charles Leclerc	16
Carlos Sainz Jr.	55
Fernando Alonso	14
Lance Stroll	18
Valentino Rossi	46
Marc Marquez	93

Driver_Internal

```
CREATE TABLE Driver_Internal (
    driver_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    driver_name VARCHAR2(100),
    country_of_origin VARCHAR2(50),
    CONSTRAINT fk_name FOREIGN KEY (driver_name)
        REFERENCES Driver(driver_name)
        ON DELETE CASCADE,
    CONSTRAINT unique_driver_name_internal UNIQUE (driver_name)
);
```

driver_id	driver_name	country_of_origin
1	Lewis Hamilton	UK
2	George Russell	UK
3	Max Verstappen	Netherlands
4	Sergio Perez	Mexico
5	Charles Leclerc	Monaco
6	Carlos Sainz Jr.	Spain
7	Fernando Alonso	Spain
8	Lance Stroll	Canada
9	Valentino Rossi	Italy
10	Marc Marquez	Spain

Racing

```
CREATE TABLE Racing (
    rs_name VARCHAR2(100) PRIMARY KEY,
    division VARCHAR2(50),
    governing_body VARCHAR2(100)
);
```

rs_name	division	governing_body
Formula 1 World Championship	Elite	FIA
MotoGP World Championship	Elite	FIM
World Endurance Championship	Endurance	FIA
Formula 2 Championship	Feeder	FIA

Racing_Series

```
CREATE TABLE Racing_Series (
    rs_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    rs_name VARCHAR2(100),
    CONSTRAINT fk_racing_series_name FOREIGN KEY (rs_name)
        REFERENCES Racing(rs_name)
        ON DELETE SET NULL,
    CONSTRAINT unique_rs_name_link UNIQUE (rs_name)
);
```

rs_id	rs_name
1	Formula 1 World Championship
2	MotoGP World Championship
3	World Endurance Championship
4	Formula 2 World Championship

Engineering_Team

```
CREATE TABLE Engineering_Team (
    eng_team_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    team_name VARCHAR2(100),
    department VARCHAR2(100),
    HQ_address VARCHAR2(100)
);
```

eng_team_id	team_name	department	HQ_address
1	Mercedes High Performance Powertrains	Engine	Brixworth, UK
2	Mercedes Aerodynamics Dept	Aerodynamics	Brackley, UK
3	Red Bull Powertrains	Engine	Milton Keynes, UK
4	Red Bull Aerodynamics Dept	Aerodynamics	Milton Keynes, UK
5	Ferrari Gestione Sportiva	Engine	Maranello, Italy
6	Ferrari Aerodynamics Dept	Aerodynamics	Maranello, Italy
7	Honda Racing Corporation	Engine	Asaka, Japan
8	Yamaha Factory Racing	Chassis	Lesmo, Italy

Engineering_Assignment

```
CREATE TABLE Engineer_Assignment (
    eng_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    name VARCHAR2(100),
    proficiency VARCHAR2(50),
    years_experience NUMBER,
    eng_team_id NUMBER,
    CONSTRAINT fk_eng_team_id FOREIGN KEY (eng_team_id)
        REFERENCES Engineering_Team(eng_team_id)
        ON DELETE SET NULL
);
```

eng_id	name	proficiency	years_experience	eng_team_id
1	Andy Cowell	Expert	20	1
2	Hywel Thomas	Senior	15	1
3	James Allison	Expert	25	2
4	Mike Elliot	Senior	18	2
5	Ben Hodgkinson	Expert	22	3
6	Adrian Newey	Legend	35	4
7	Pierre Wache	Senior	16	4
8	Rory McIlroy	Junior	5	4
9	Enrico Gualtieri	Expert	19	5
10	David Sanchez	Senior	14	6
11	Sinichi Kokubu	Expert	28	7
12	Massimo Meregalli	Senior	20	8

Vehicle

```
CREATE TABLE Vehicle (
    vehicle_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    model VARCHAR2(100),
    year_first_produced NUMBER(4) CHECK (year_first_produced BETWEEN 1000 AND 2015),
    team_id NUMBER,
    driver_id NUMBER,
    CONSTRAINT fk_team FOREIGN KEY (team_id)
        REFERENCES Team(team_id)
        ON DELETE SET NULL,
    CONSTRAINT fk_driver FOREIGN KEY (driver_id)
        REFERENCES Driver_Internal(driver_id)
        ON DELETE SET NULL
);
```

vehicle_id	model	year_first_produced	team_id	driver_id
1	Mercedes-AMG F1 W14	2014	1	1
2	Mercedes-AMG F1 W14	2014	1	2
3	Red Bull RB19	2013	2	3
4	Red Bull RB19	2013	2	4
5	Ferrari SF-23	2015	3	5
6	Ferrari SF-23	2015	3	6
7	Aston Martin AMR23	2012	4	7
8	Aston Martin AMR23	2012	4	8
9	Yamaha YZR-M1	2002	3	9
10	Honda RC213V	2012	2	10
11	Ducati Desmosedici GP23	2015	3	NULL

Car

```
CREATE TABLE Car (
    vehicle_id NUMBER PRIMARY KEY,
    drivetrain CHAR(3) CHECK (drivetrain IN ('FWD', 'RWD', '4WD', 'AWD')),
    car_type VARCHAR2(50),
    CONSTRAINT fk_car_vehicle FOREIGN KEY (vehicle_id)
        REFERENCES Vehicle(vehicle_id)
        ON DELETE CASCADE
);
```

vehicle_id	drivetrain	car_type
1	RWD	Formula 1
2	RWD	Formula 1
3	RWD	Formula 1
4	RWD	Formula 1
5	RWD	Formula 1
6	RWD	Formula 1
7	RWD	Formula 1
8	RWD	Formula 1

University of British Columbia, Vancouver
Department of Computer Science

Motorcycle

```
CREATE TABLE Motorcycle (
    vehicle_id NUMBER PRIMARY KEY,
    engine_cc NUMBER CHECK (engine_cc BETWEEN 100 AND 3000),
    motorcycle_type VARCHAR2(50),
    CONSTRAINT fk_motorcycle_vehicle FOREIGN KEY (vehicle_id)
        REFERENCES Vehicle(vehicle_id)
        ON DELETE CASCADE
);
```

vehicle_id	engine_cc	motorcycle_type
9	1000	MotoGP
10	1000	MotoGP
11	1000	MotoGP

Project

```
CREATE TABLE Project (
    project_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    upgrade VARCHAR2(50),
    duration NUMBER,
    status VARCHAR2(50),
    vehicle_id NUMBER NOT NULL,
    eng_team_id NUMBER,
    CONSTRAINT fk_vehicle_id FOREIGN KEY (vehicle_id)
        REFERENCES Vehicle(vehicle_id)
        ON DELETE CASCADE,
    CONSTRAINT fk_project_eng_team FOREIGN KEY (eng_team_id)
        REFERENCES Engineering_Team(eng_team_id)
        ON DELETE SET NULL
);
```

project_id	upgrade	duration	status	vehicle_id	eng_team_id
1	Front Wing Redesign	45	Completed	1	2
2	Floor Edge Update	30	In Progress	3	4
3	ERS Power Output Increase	90	Testing	5	5
4	Swingarm Stiffness Mod	60	Completed	9	8
5	ICE Reliability Fix	20	Completed	1	1

Upgrade

```
CREATE TABLE Upgrade (
    brand VARCHAR2(50),
    part_number NUMBER,
    project_id NUMBER,
    status VARCHAR2(50),
    PRIMARY KEY (brand, part_number, project_id),
    CONSTRAINT fk_project_id FOREIGN KEY (project_id)
        REFERENCES Project(project_id)
        ON DELETE CASCADE
);
```

brand	part_number	project_id	status
MercAero	1401	1	Installed
MercAero	1402	1	Installed
RBAero	1905	2	Fabricating
RBAero	1906	2	Pending
FerrariEng	2310	3	Bench Tested
YamahaFac	1001	4	Installed
MercEng	1455	5	Installed

Funds

```
CREATE TABLE Funds (
    sponsor_id NUMBER,
    team_id NUMBER,
    PRIMARY KEY (sponsor_id, team_id),
    contract_start_date DATE,
    contract_end_date DATE,
    CONSTRAINT fk_sponsor_id FOREIGN KEY (sponsor_id)
        REFERENCES Sponsor(sponsor_id)
        ON DELETE CASCADE,
    CONSTRAINT fk_funds_team_id FOREIGN KEY (team_id)
        REFERENCES Team(team_id)
        ON DELETE CASCADE
);
```

sponsor_id	team_id	contract_start_date	contract_end_date
1	1	2022-01-01	2026-12-31
2	1	2020-01-01	2024-12-31
3	2	2023-01-01	2027-12-31
4	2	2016-01-01	2025-12-31
5	3	1996-01-01	2028-12-31
6	3	2022-01-01	2024-12-31
4	3	2024-01-01	2026-12-31
7	4	2022-01-01	2027-12-31
8	4	2021-01-01	2025-12-31

Competes_In

```
CREATE TABLE Competes_In (
    team_id NUMBER,
    racing_series_id NUMBER,
    PRIMARY KEY (team_id, racing_series_id),
    ranking_in_series NUMBER,
    CONSTRAINT fk_competes_team_id FOREIGN KEY (team_id)
        REFERENCES Team(team_id)
        ON DELETE CASCADE,
    CONSTRAINT fk_rs_id FOREIGN KEY (racing_series_id)
        REFERENCES Racing_Series(rs_id)
        ON DELETE CASCADE
);
```

team_id	racing_series_id	ranking_in_series
1	1	3
2	1	1
3	1	2
4	1	4
3	2	1
2	2	2
3	3	1
1	4	5
3	4	2
2	3	2
2	4	3

University of British Columbia, Vancouver
Department of Computer Science

Rubric Satisfying SQL Queries

Query 1: Insert

```
INSERT INTO Engineer_Assignment (name, proficiency, years_experience, eng_team_id)
            VALUES (:name, :proficiency, :yearsExperience, :engTeamId)
```

File: backend/services/entities/engineerService.js
Line(s): 19-21

Query 2: Update

```
UPDATE Engineer_Assignment
            SET ${updates.join(", ")}
            WHERE name = :oldName
```

File: backend/services/entities/engineerService.js
Line(s): 176-178

Since Oracle does not support the ON UPDATE constraint, we acknowledge that foreign keys should be updated but have chosen a relation to update where the foreign key does not need to be updated.

Query 3: Delete

```
DELETE FROM Sponsor_Tier WHERE tier_level = :tierLevel
```

File: backend/services/entities/sponsorService.js
Line(s): 317

The ON DELETE CASCADE is implemented in the SQL DDL statement Create Table for Sponsor where all associated sponsors are deleted when a sponsor tier is deleted

```
CREATE TABLE Sponsor(
                        sponsor_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
                        sponsor_name VARCHAR2(50) NOT NULL,
                        tier_level VARCHAR2(50),
                        point_of_contact VARCHAR2(100),
                        CONSTRAINT unique_sponsor_name UNIQUE (sponsor_name),
                        CONSTRAINT fk_sponsor_tier FOREIGN KEY (tier_level)
                          REFERENCES Sponsor_Tier(tier_level)
                          ON DELETE CASCADE
        )
```

File: backend/services/entities/sponsorService.js
Line(s): 58-67

Query 4: Selection

```
SELECT ea.eng_id, ea.name, ea.proficiency, ea.years_experience, et.team_name
        FROM Engineer_Assignment ea
        JOIN Engineering_Team et ON ea.eng_team_id = et.eng_team_id
        WHERE
```

File: backend/services/entities/engineerService.js
Line(s): 326-329

Query 5: Projection

```
SELECT ${sqlColumns.join(", ")}
      FROM Engineer_Assignment ea
      JOIN Engineering_Team et ON ea.eng_team_id = et.eng_team_id
```

File: backend/services/entities/engineerService.js
Line(s): 301-303

Query 6: Join

```
SELECT ea.name, ea.proficiency, ea.years_experience, et.team_name, et.department, et.HQ_address
        FROM Engineer_Assignment ea
        JOIN Engineering_Team et ON ea.eng_team_id = et.eng_team_id
        WHERE LOWER(et.department) LIKE LOWER(:department)
```

File: backend/services/entities/engineerService.js
Line(s): 401-404

Query 7: Aggregation with GROUP BY

```
SELECT ET.team_name, COUNT(EA.eng_id) AS num_engineers
          FROM Engineering_Team ET
          JOIN Engineer_Assignment EA ON ET.eng_team_id = EA.eng_team_id
          GROUP BY ET.team_name
```

File: backend/services/queries/aggregateService.js
Line(s): 6-9

Function: Returns the number of engineers per team

Query 8: Aggregation with HAVING

```
SELECT ET.team_name, COUNT(EA.eng_id) AS num_engineers
        FROM Engineering_Team ET
        JOIN Engineer_Assignment EA ON ET.eng_team_id = EA.eng_team_id
        GROUP BY ET.team_name
        HAVING COUNT(EA.eng_id) > 1
```

File: backend/services/queries/aggregateService.js
Line(s): 17-21

Function: Returns the names and engineer count of engineering teams that have more than one engineer

Query 9: Nested Aggregation with GROUP BY

```
SELECT eng_team_id, AVG(years_experience) AS avg_experience
        FROM Engineer_Assignment
        GROUP BY eng_team_id
        HAVING AVG(years_experience) > (SELECT AVG(years_experience) FROM Engineer_Assignment)
```

File: backend/services/queries/aggregateService.js
Line(s): 30-33

Function: Returns the identification number and average years of experience of engineering teams whose average years of experience across the engineers is greater than the average years of experience of all engineers

Query 10: Division

```
SELECT T.team_id
        FROM Team T
        WHERE NOT EXISTS (SELECT RS.rs_id
                FROM Racing_Series RS
                MINUS
                SELECT CI.racing_series_id
                FROM Competes_In CI
                WHERE CI.team_id = T.team_id)
```

File: backend/services/queries/aggregateService.js
Line(s): 42-49

Function: Returns the identification number of teams that participate in all racing series.