CPSC 304 Project Cover Page

Milestone #: 4

Date: Fri, Nov 25, 2022 **Group Number:** 11

Name	Student Number	CS Alias (UserID)	Preferred Email Address
Kyle Rich	36316933	h7w2nb	kyledvrich@gmail.com
Eric Liu	95724787	p3v2b	liu.eric103@gmail.com
Anisha Gill	27909993	r4h0d	gill.anisha@outlook.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.

Project Description

Our project aims to capture all the information necessary for the Federation Internationale de l'Automobile (FIA) – the Formula 1 league – to collect and reference data from current or previous seasons. This includes everything, from drivers, teams, engineers, managers, races, grand prix, circuits, and so on anything the FIA could possibly need to manage the sport.

Our design allows a Formula 1 sport manager to access quick, useful information about the top performing teams, players, etc. It also allows them to filter and view all data within the database, such as about drivers, races, teams, etc. They also have the power to access a Race Details dashboard where they can quickly filter joined tables (races, raced, and drivers), so they can quickly figure out important information about the winners of each race, etc. Additionally, they can edit the details of drivers, add new circuits, and delete historic season data if necessary.

Overall, our project has the goal of providing an Formula 1 sport manager with all the information they could need at their fingertips.

Schema Differences

We spent considerable time in milestone 2 to make sure our schema was mostly 100% perfect, to minimize our work later on. Therefore, we actually made no changes to our schema AT ALL. The schema's listed below in the "Schemas and Screenshots" section are **literally** copy and pasted from milestone 2. We realize there is a 2 point mark for this section, but we do not think it is fair to lose that mark simply for being proactive, and making sure our schema was correct the first time around during milestone 2.

The only changes we made were insignificant:

- 1) using "_" instead of UpperCamelCase or lowerCamelCase
- 2) using singular, rather than plural (for example, "car" rather than "cars")
- 3) fixing typos
- 4) re-ordering some of the data fields. (for example, on the drivers table, fastest_laps now comes before podiums)

Schema and Screenshots

Cars: Cars(carNumber: int, teamName: string, powerUnit: string, chassisModel: string, toDate: Date, fromDate: Date,

driverID: int)

	number [PK] integer	team_name [PK] character varying (255)	power_unit character varying (255)	chassis_model character varying (255)	to_date date	from_date date	driver_id integer
1	1	Mercedes-AMG Petronas Formula 1 Team	Renault	MD34s	2019-04-10	2022-10-14	149
2	2	Mercedes-AMG Petronas Formula 1 Team	Renault	MD34s	2019-04-10	2022-10-14	148
3	2	Red Bull Racing	Honda	RB2d7	2018-12-11	2020-10-15	240
4	3	Scuderia Ferrari	Ferrari	Frh9d	2013-05-12	2020-10-16	241
5	4	McLaren F1 Team	Mercedes	MLnw0	2020-04-13	2020-10-17	132
6	5	Alpine F1 Team	Renault	APjf8	2021-04-14	2020-10-18	164
7	6	Red Bull Racing	Honda	RB2d7	2018-12-11	2020-10-15	239
8	7	McLaren F1 Team	Mercedes	MLnw0	2020-04-13	2020-10-17	131
9	8	Scuderia Ferrari	Ferrari	Frh9d	2013-05-12	2020-10-16	242
10	9	Alpine F1 Team	Renault	APjf8	2021-04-14	2020-10-18	163

Circuits: Circuits(<u>circuitName: string</u>, mapURL: string, lastLengthIUsed: int, direction: string, type: string, changeInElevation: int, **grandPrixTitle: string**)

	name [PK] character varying (255)	map_url character varying (255)	last_length_used /	direction character varying (255)	type character varying (255)	change_in_elevation integer	grand_prix_title character varying (255)
1	Bahrain International Circuit	http://en.wikipedia.org/wiki/Bahrain_International_Circuit	4	cw	race track	63	Bahrain Grand Prix
2	Circuit de Barcelona-Catalunya	http://en.wikipedia.org/wiki/Circuit_de_Barcelona-Catalun	5	clockwise	race track	45	Catalunya Grand Prix
3	Circuit de Monaco	http://en.wikipedia.org/wiki/Circuit_de_Monaco	6	counterclockwise	street track	42	Monaco Grand Prix
4	Circuit Gilles Villeneuve	http://en.wikipedia.org/wiki/Circuit_Gilles_Villeneuve	6	counterclockwise	race track	20	Canadian Grand Prix
5	Istanbul Park	http://en.wikipedia.org/wiki/Istanbul_Park	5	clockwise	race track	31	Istanbul Grand Prix

Drivers: Drivers(<u>driverID: int,</u> name: string, nationality: string, dateOfBirth: Date, placeOfBirth: string, worldChampionships: int, fastestLaps: int, racesWon: int, podiums: int)

	id [PK] integer	name character varying (255)	nationality character varying (255)	dob date	place_of_birth character varying (255)	world_championships integer	fastest_laps /	races_won integer	podiums integer
1	131	Daniel Ricciardo	Australian	1989-07-01	Australia	0	16	8	32
2	132	Lando Norris	British	1981-07-29	UK	0	51	7	126
3	148	George Russell	British	1998-02-15	UK	0	0	3	13
4	149	Lewis Hamilton	British	1985-01-07	UK	7	15	5	285
5	163	Fernando Alonso	Spanish	1981-07-29	Spain	2	23	32	98
6	164	Esteban Ocon	French	1981-10-19	France	0	68	2	69
7	239	Sergio Pérez	Mexican	1990-05-10	Mexio	0	3	4	26
8	240	Max Verstappen	Dutch	1977-05-10	Netherlands	1	39	7	289
9	241	Carlos Sainz Jr.	Argentine	1985-06-27	Spain	0	65	2	213
10	242	Charles Leclerc	Monégasque	1997-10-16	Monaco	0	24	5	24

DrivesOn: DrivesOn(<u>teamName: string</u>, <u>driverID: int</u>, fromDate: Date, toDate: Date)

	team_name [PK] character varying (255)	driver_id [PK] integer	from_date /	to_date date
1	Alpine F1 Team	163	2019-12-15	[null]
2	Alpine F1 Team	164	2018-12-15	[null]
3	McLaren F1 Team	131	2018-07-01	[null]
4	McLaren F1 Team	132	2014-07-01	[null]
5	Mercedes-AMG Petronas Formula 1 Team	148	2019-06-15	[null]
6	Mercedes-AMG Petronas Formula 1 Team	149	2015-06-15	[null]
7	Red Bull Racing	239	2018-07-04	[null]
8	Red Bull Racing	240	2019-04-13	[null]
9	Scuderia Ferrari	241	2019-09-28	[null]
10	Scuderia Ferrari	242	2018-09-28	[null]

DroveDuring: DroveDuring(<u>driverID: int, seasonYear: int</u>, driverRank: int)

	driver_id [PK] integer	season_year [PK] integer	driver_rank integer
1	131	2020	7
2	132	2020	4
3	148	2020	9
4	149	2020	1
5	163	2020	12
6	164	2020	16
7	239	2020	14
8	240	2020	2
9	241	2020	6
10	242	2020	5

Engineers: Engineers(<u>engineerID: int</u>, name: string, nationality: string, dateOfBirth: Date, placeOfBirth: string, educationLevel: string)

	id [PK] integer	name character varying (255)	nationality character varying (255)	dob date	place_of_birth character varying (255)	education_level character varying (255)
1	1	David	Honduran	1995-05-22	Honduras	Bachelors or equivalent level
2	7	Jenson	Cuban	1994-11-16	Canada	Bachelor's or equivalent level
3	11	Mellisa	Haitian	1993-10-05	United States of America	Master's or equivalent level
4	14	Jarno	Thai	1983-04-13	Canada	Post-secondary non-tertiary education
5	16	Mark	Mongolian	1987-08-10	Philippines	Bachelor's or equivalent level

EngineersFor: EngineersFor(<u>teamName: string</u>, <u>engineerID: int</u>, fromDate: Date, toDate: Date)

	team_name [PK] character varying (255)	engineer_id [PK] integer	from_date /	to_date /
1	Alpine F1 Team	7	2017-07-14	[null]
2	McLaren F1 Team	16	2020-01-29	[null]
3	Mercedes-AMG Petronas Formula 1 Team	1	2016-10-28	[null]
4	Red Bull Racing	14	2019-11-05	[null]
5	Scuderia Ferrari	11	2017-06-12	[null]

GrandPrix: GrandPrix(<u>title: string</u>, country: string)

	title [PK] character varying (255)	country character varying (255)
1	Bahrain Grand Prix	Bahrain
2	Canadian Grand Prix	Canada
3	Catalunya Grand Prix	Spain
4	Istanbul Grand Prix	Turkey
5	Monaco Grand Prix	Monaco

Investors: Investors(name: string)

	name [PK] character varying (255)
1	AMD
2	Chrome
3	Dell
4	Goldman Sachs
5	Hewlett Packard Enterprise

InvestsIn: InvestIn(investorName, teamName, startDate, endDate, amount)

	investor_name [PK] character varying (255)	team_name [PK] character varying (255)	start_date /	end_date /	amount /
1	AMD	Mercedes-AMG Petronas Formula 1 Team	2015-06-15	[null]	23145731
2	Chrome	McLaren F1 Team	2020-09-28	[null]	21050104
3	Dell	McLaren F1 Team	2014-07-01	[null]	16632288
4	Goldman Sachs	McLaren F1 Team	2018-12-15	[null]	8282840
5	Hewlett Packard Enterprise	Mercedes-AMG Petronas Formula 1 Team	2020-04-13	[null]	17351333

InvestsInAmounts: InvestInAmounts(amount, level)

	amount [PK] integer	level character varying (255)
1	8282840	bronze
2	16632288	silver
3	17351333	silver
4	21050104	gold
5	23145731	platinum

Maintains: Maintains(<u>carNumber: int</u>, <u>teamName: string</u>, <u>pitCrewID: int</u>)

	car_number [PK] integer	team_name [PK] character varying (255)	pit_crew_id [PK] integer
1	1	Mercedes-AMG Petronas Formula 1 Team	22
2	2	Red Bull Racing	22
3	3	Scuderia Ferrari	27
4	4	McLaren F1 Team	21
5	5	Alpine F1 Team	35
6	6	Red Bull Racing	22
7	7	McLaren F1 Team	21
8	8	Scuderia Ferrari	27
9	9	Alpine F1 Team	35

MaintainsCarsFor: MaintainsCarsFor(<u>teamName: string</u>, <u>pitCrewID: int</u>, fromDate: Date, toDate: Date)

	team_name [PK] character varying (255)	pit_crew_id [PK] integer	from_date /	to_date /
1	Alpine F1 Team	35	2019-06-26	[null]
2	McLaren F1 Team	21	2016-12-17	[null]
3	Mercedes-AMG Petronas Formula 1 Team	22	2016-11-23	[null]
4	Red Bull Racing	21	2020-03-25	[null]
5	Scuderia Ferrari	27	2019-01-10	[null]

Managers: Managers(managerID: int, name: string, nationality: string, dateOfBirth: Date, placeOfBirth: string, role: string)

	id [PK] integer	name character varying (255)	nationality character varying (255)	place_of_birth character varying (255)	dob date	role character varying (255)
1	275	Gianmaria	Italian	Italy	1981-12-15	principal
2	320	Antônio	French	France	1983-10-15	principal
3	322	Olivier	British	Britan	1993-12-15	principal
4	341	Giorgio	Italian	Italy	1982-12-15	principal
5	356	Cristiano	French	France	1993-08-16	principal

Manages : Manages(<u>teamName: string</u>, <u>managerID: int</u>, fromDate: Date, toDate: Date)

	manager_id [PK] integer	team_name [PK] character varying (255)	from_date /	to_date /
1	275	Alpine F1 Team	2018-12-15	[null]
2	320	Mercedes-AMG Petronas Formula 1 Team	2020-10-15	[null]
3	322	Scuderia Ferrari	2018-12-15	[null]
4	341	McLaren F1 Team	2019-12-15	[null]
5	356	Red Bull Racing	2013-08-16	[null]

ParticipatedIn: ParticipatedIn(<u>teamName: string, seasonYear: int</u>, constructorRank: int)

	team_name [PK] character varying (255)	season_year [PK] integer	constructor_rank integer
1	Alpine F1 Team	2020	9
2	McLaren F1 Team	2020	4
3	Mercedes-AMG Petronas Formula 1 Team	2020	2
4	Red Bull Racing	2020	1
5	Scuderia Ferrari	2020	7

PitCrew: PitCrew(pitCrewID: int, Name: string, nationality: string, dateOfBirth: Date, placeOfBirth: string, position: string, bestTime: Time)

	id [PK] integer	name character varying (255)	nationality character varying (255)	dob date	place_of_birth character varying (255)	position character varying (255)	best_time time without time zone
1	21	Sam	Iranian	1999-07-19	Iran	BR	00:00:01.82
2	22	Samantha	German	1989-12-01	Germany	BR	00:00:02.36
3	23	Alex	Spanish	1986-01-01	Spain	FR	00:00:01.97
4	27	Lilly	Bulgarian	1990-03-10	United States of America	FL	00:00:02.2
5	35	Heikki	Swedish	1984-03-22	Sweden	BL	00:00:01.88

Races: Races(<u>raceID</u>: <u>int</u>, raceDate: Date, temperature: int, fastestLapTime: Time, redFlagCount: int, **safetyCarModel**: string, DNFCount: int, poleSitterDriverID: int, seasonYear: int, circuitName: string)

	id [PK] integer	date /	temperature /	fastest_lap time without time zone	safety_car_model character varying (255)	dnf_count /	pole_sitter_id /	season_year /	circuit_name character varying (255)	red_flag_count integer	•
1	400	2020-07-21	10	00:00:01.4	Porsche 914	2	149	2020	Bahrain International Circuit		1
2	401	2022-10-18	10	00:00:02.3	Lamborghini Countach	6	149	2020	Circuit de Barcelona-Catalunya		6
3	402	2020-12-17	17	00:00:01.5	Mercedes-AMG GT Black Series	3	241	2020	Istanbul Park		2
4	403	2020-10-15	27	00:00:01.9	Lamborghini Diablo	7	149	2020	Circuit de Monaco		4
5	404	2020-07-25	21	00:00:03.4	Mercedes-AMG GT R	4	164	2020	Circuit Gilles Villeneuve		3

Raced: Raced(<u>raceID: int</u>, <u>driverID: int,</u> placement: int)

	race_id [PK] integer	driver_id [PK] integer	placement integer
1	400	149	1
2	400	240	2
3	401	148	3
4	401	149	2
5	401	239	10
6	401	240	1
7	402	149	4
8	402	240	5
9	402	241	3
10	402	242	1
11	403	131	4
12	403	132	5
13	403	149	1
14	403	240	2
15	404	149	3
16	404	163	7
17	404	164	8
18	404	240	9

SafetyCar: SafetyCar(<u>model: string</u>, topSpeed: int)

	model [PK] character varying (255)	top_speed /
1	Lamborghini Countach	172
2	Lamborghini Diablo	201
3	Mercedes-AMG GT Black Series	202
4	Mercedes-AMG GT R	224
5	Porsche 914	175

Season: Season(<u>year: int</u>, races: int)

	year [PK] integer	races integer	,
1	2016		21
2	2017		20
3	2018		21
4	2019		21
5	2020		17

Teams: Teams(<u>teamName: string</u>, baseLocation: string, owner: string, firstTeamEntry: Date, fastestLaps: int, worldChampionships: int)

	name [PK] character varying (255)	base_location character varying (255)	owner character varying (255)	first_team_entry /	fastest_laps integer	world_championships integer
1	Alpine F1 Team	Italian	Miller	1988-11-27	4	12
2	McLaren F1 Team	French	Jones	1994-02-17	7	5
3	Mercedes-AMG Petronas Formula 1 Team	British	Johnson	1991-06-19	7	7
4	Red Bull Racing	German	Williams	1999-01-01	9	11
5	Scuderia Ferrari	British	Brown	1982-08-11	12	2

$\textbf{WorksOn}(\underline{\textbf{carNumber: int}}, \underline{\textbf{teamName: string}}, \underline{\textbf{engineerID: int}})$

	car_number [PK] integer	team_name [PK] character varying (255)	engineer_id [PK] integer
1	1	Mercedes-AMG Petronas Formula 1 Team	1
2	2	Red Bull Racing	14
3	3	Scuderia Ferrari	11
4	4	McLaren F1 Team	16
5	5	Alpine F1 Team	7
6	6	Red Bull Racing	14
7	7	McLaren F1 Team	16
8	8	Scuderia Ferrari	11
9	9	Alpine F1 Team	7

INSERT QUERY DETAILS

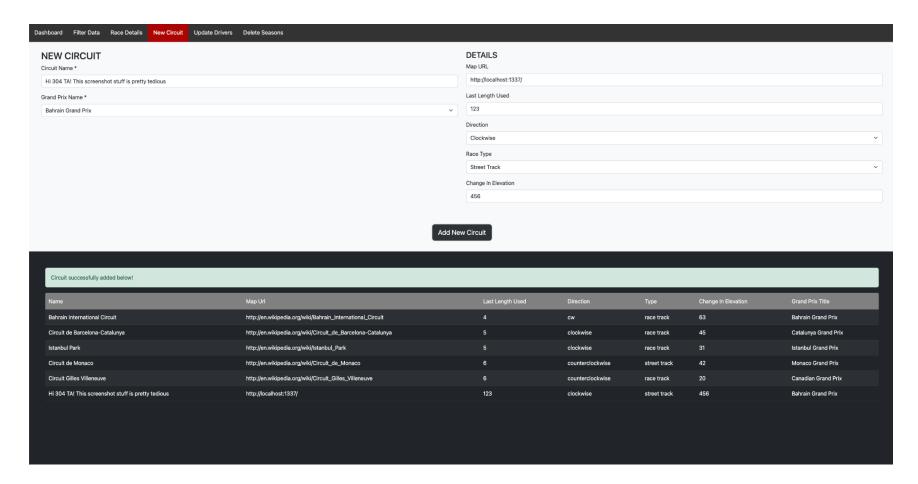
Code Reference(s)

src/server/src/rest/Server.ts::107 src/server/src/rest/interfaces.ts::16 src/server/src/QueryBuilder.ts::69 src/public/pages/newCircuit.js::10

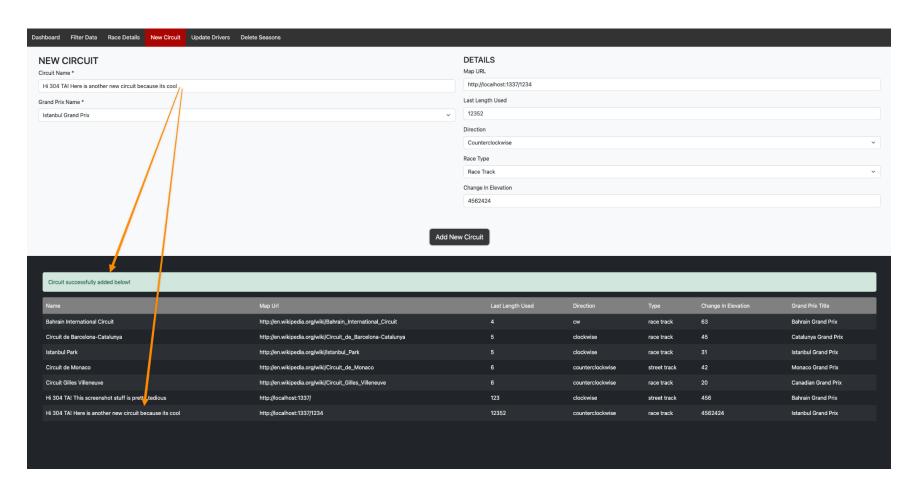
Query Reference (These queries are built dynamically according to infinite number of user filters)

```
static buildInsertQuery(insertQuery: InsertQuery): string {
  let query = "INSERT INTO circuit VALUES (" + "'" + insertQuery.columnValues.name + "'";
  insertQuery.columnValues.map_url ? query += ",'" + insertQuery.columnValues.map_url + "'" : query += "," + "Null";
  insertQuery.columnValues.last_length_used ? query += "," + insertQuery.columnValues.last_length_used : query += "," + "Null"
  insertQuery.columnValues.direction ? query += ",'" + insertQuery.columnValues.direction + "'" : query += "," + "Null";
  insertQuery.columnValues.type ? query += ",'" + insertQuery.columnValues.type + "'" : query += "," + "Null";
  insertQuery.columnValues.change_in_elevation ? query += "," + insertQuery.columnValues.change_in_elevation : query += "," +"Null";
  query += ",'" + insertQuery.columnValues.grand_prix_title + "')"
  return query;
}
```

Before INSERT Query



After INSERT Query



DELETE QUERY DETAILS

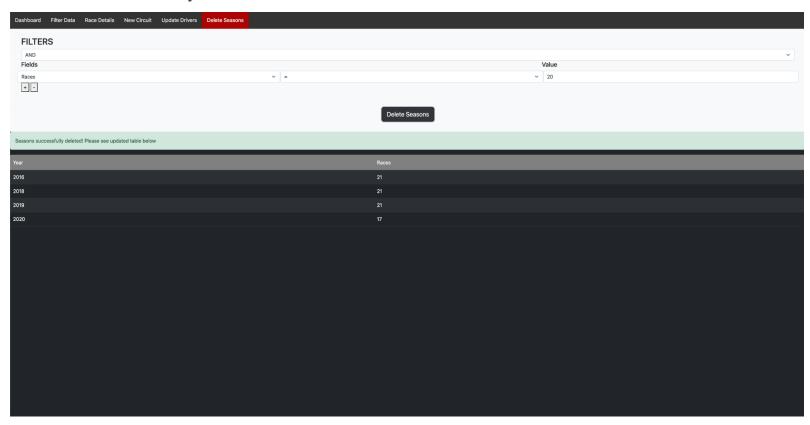
Code Reference(s)

/Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/Server.ts::133 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/QueryBuilder.ts::80 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/interfaces.ts::44 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/public/pages/deleteSeasons.js::11

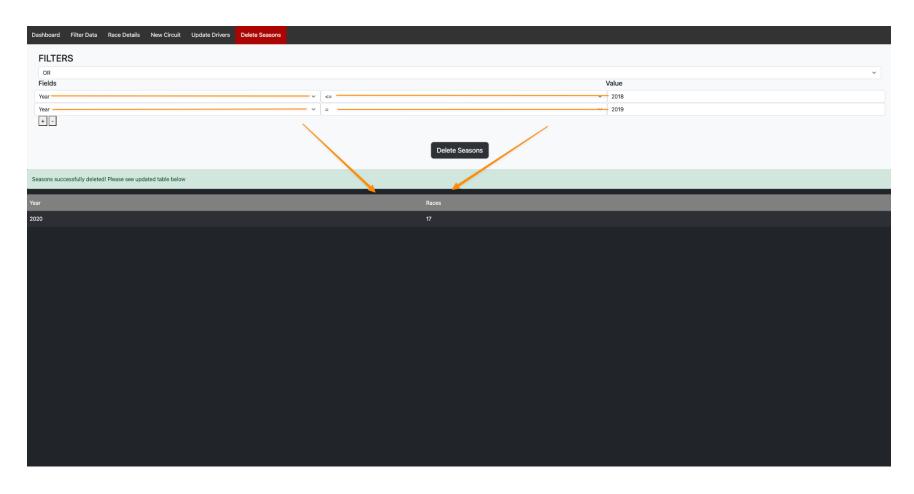
Query Reference (These queries are built dynamically according to infinite number of user filters)

```
static buildDeleteQuery(deleteQuery: DeleteQuery): string {
    let query = 'DELETE FROM ';
    query += deleteQuery.tableName;
    if (deleteQuery.filter) {
        if(deleteQuery.filter.filterItems.length === 0) {throw new Error("Please enter numbers for any filter items")}
        query += ' WHERE ';
        const logicalOperator = deleteQuery.filter.type === 'and' ? ' AND ' : ' OR ';
        const deleteItems = deleteQuery.filter.filterItems;
        for (let i = 0; i < deleteItems.length; i++) {
            if(deleteItems[i].value == "") {throw new Error("filter items are invalid")}
            query += deleteItems[i].column + deleteItems[i].compareOperator + deleteItems[i].value;
            if (i != deleteItems.length - 1) query += logicalOperator;
        }
    }
    return query;
}</pre>
```

Before DELETE Query



After DELETE Query



UPDATE QUERY DETAILS

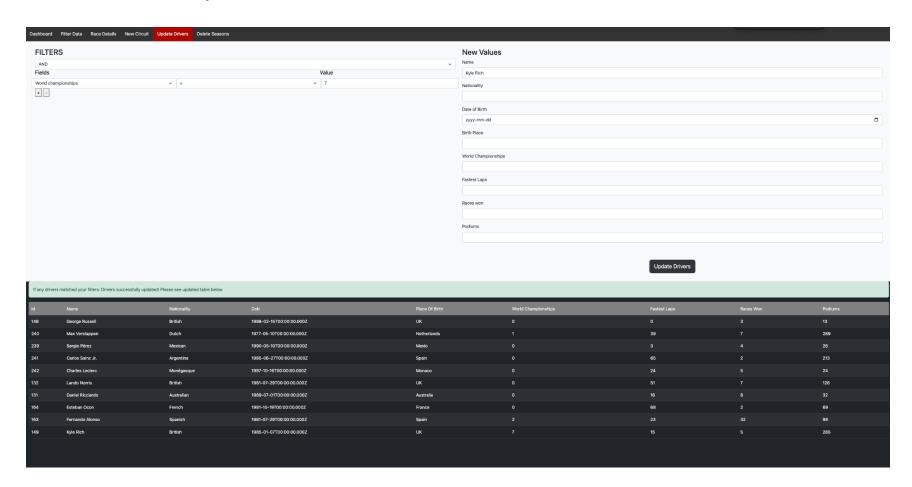
Code Reference(s)

/Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/QueryBuilder.ts::97 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/public/pages/updateDrivers.js::9 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/interfaces.ts::29 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/Server.ts::148

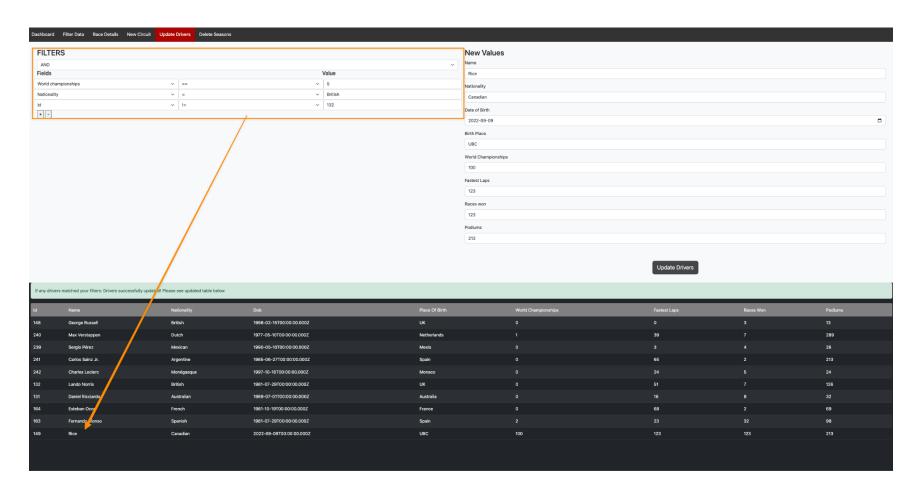
Query Reference (These queries are built dynamically according to infinite number of user filters)

```
static buildUpdateQuery(updateQuery: UpdateQuery):                           string {
 let query = 'UPDATE ';
 query += updateQuery.tableName;
 query += ' SET ';
 if (Object.keys(updateQuery.newColumnValues).length === 0) {
   throw new Error('Please specify at least one column to update');
 let temp_query = '';
 for (const col in updateQuery.newColumnValues) {
   const value = String(updateQuery.newColumnValues[col as keyof typeof updateQuery.newColumnValues]);
   query += col + ' = ' + `'` + value + `'` + ', ';
 temp_query = query.slice(0,-2);
 if (updateQuery.filter) {
   temp_query += ' WHERE ';
   const logicalOperator = updateQuery.filter.type === 'and' ? ' AND ' : ' OR ';
   const filterItems = updateQuery.filter.filterItems;
   for (let i = 0; i < filterItems.length; i++) {</pre>
     if(filterItems[i].value == "") {throw new Error("filter items are invalid")}
     query += filterItems[i].column + filterItems[i].compareOperator + filterItems[i].value;
     temp_query += filterItems[i].column + ' ' + filterItems[i].compareOperator + ' ' + filterItems[i].value;
     if (i != filterItems.length - 1) temp_query += logicalOperator;
 return temp_query;
```

Before UPDATE Query



After UPDATE Query



SELECTION / PROJECTION QUERY DETAILS

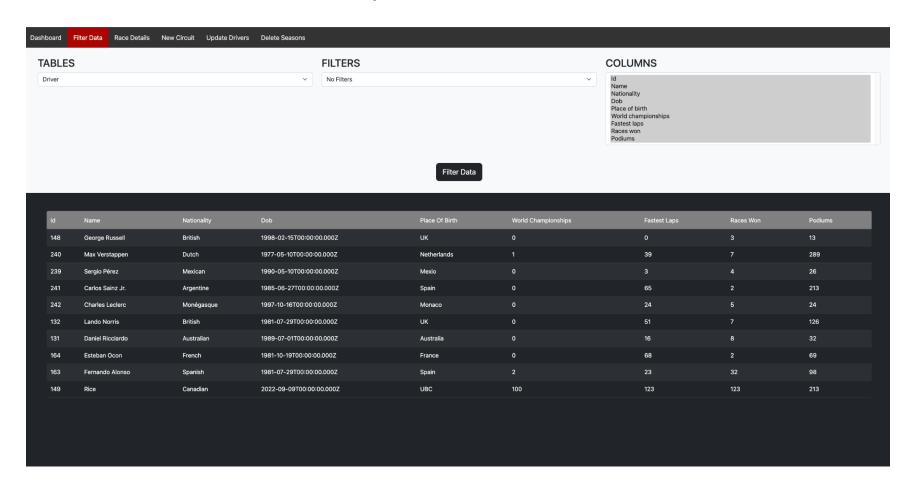
Code Reference(s)

/Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/QueryBuilder.ts::8 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/Server.ts::77 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/interfaces.ts::1 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/public/pages/filterData.js::12

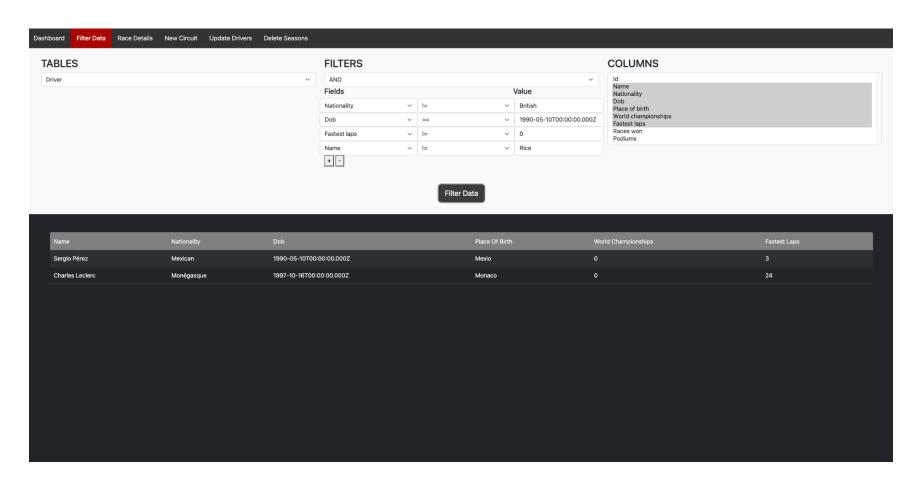
Query Reference (These queries are built dynamically according to infinite number of user filters)

```
static buildFilterQuery(filterQuery: FilterQuery): string {
  let query = 'SELECT ';
 for (let i = 0; i < filterQuery.selectedColumns.length; i++) {</pre>
    query += filterQuery.selectedColumns[i];
   if (i != filterQuery.selectedColumns.length - 1) guery += ', ';
  if (filterQuery.selectedColumns.length === 0) {
    throw new Error('Please select at least one column');
  query += ' FROM ';
  query += filterQuery.tableName;
  if (filterQuery.filter) {
    query += ' WHERE ';
    const logicalOperator = filterQuery.filter.type === 'and' ? ' AND ' : ' OR ';
    const filterItems = filterQuery.filter.filterItems;
    for (let i = 0; i < filterItems.length; i++) {</pre>
      if(!filterItems[i].value) {
        throw new Error("filter items are invalid")
      query += filterItems[i].column + ' ' + filterItems[i].compareOperator + ' ' + filterItems[i].value;
      if (i != filterItems.length - 1) query += logicalOperator;
```

Before SELECTION / PROJECTION Query



After SELECTION / PROJECTION Query



JOIN QUERY DETAILS

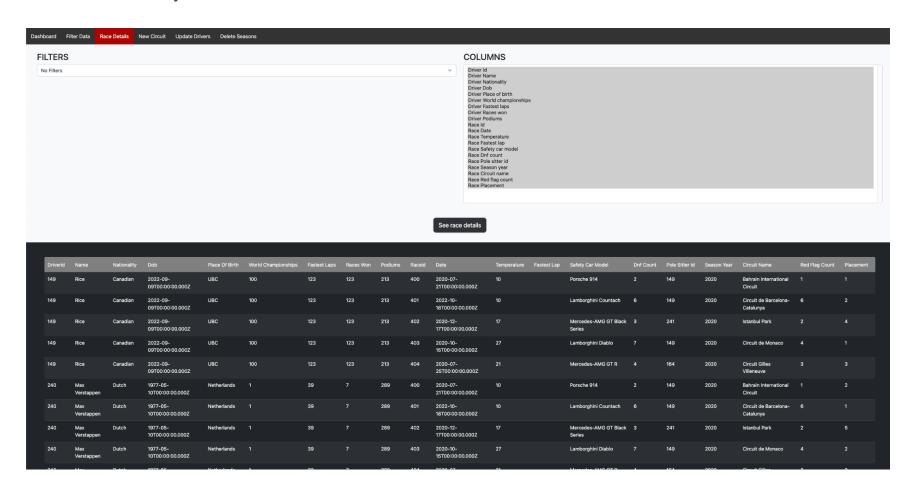
Code Reference(s)

/Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/QueryBuilder.ts::35 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/Server.ts::94 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/interfaces.ts::7 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/public/pages/raceDetails.js::13

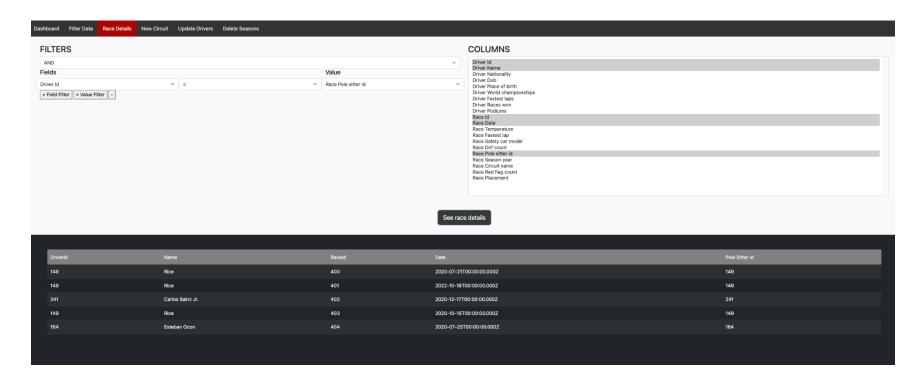
Query Reference (These queries are built dynamically according to infinite number of user filters)

```
static buildJoinQuery(joinQuery: JoinQuery): string {
  let query = 'SELECT ';
  for (let i = 0; i < joinQuery.selectedColumns.length; i++) {</pre>
   query += joinQuery.selectedColumns[i];
    if (i != joinQuery.selectedColumns.length - 1) query += ', ';
  if (joinQuery.selectedColumns.length === 0) {
  query += ' FROM ' + joinQuery.tableName1 + ', ' + joinQuery.tableName2 + ', ' + joinQuery.tableName3;
  query += ' WHERE ' + joinQuery.tableName1 + '.id = ' + joinQuery.tableName3 + '.' + joinQuery.tableName1 + '_id'
  query += ' AND ' + joinQuery.tableName3 + '.' + joinQuery.tableName2 + '_id = ' + joinQuery.tableName2 + '.id';
  if (joinQuery.filter && joinQuery.filter.filterItems.length > 0) {
   query += ' AND (';
  if (joinQuery.filter && joinQuery.filter.filterItems.length > 0) {
   const logicalOperator = joinQuery.filter.type === 'and' ? ' AND ' : ' OR ';
    const filterItems = joinQuery.filter.filterItems;
   for (let i = 0; i < filterItems.length; i++) {</pre>
      if(!filterItems[i].value) {
        throw new Error("filter items are invalid")
      query += filterItems[i].column + ' ' + filterItems[i].compareOperator + ' ' + filterItems[i].value;
      if (i != filterItems.length - 1) query += logicalOperator;
  if (joinQuery.filter && joinQuery.filter.filterItems.length > 0) {
    query += ')';
  return query;
```

Before JOIN Query



After JOIN Query



Aggregation with Group By DETAILS

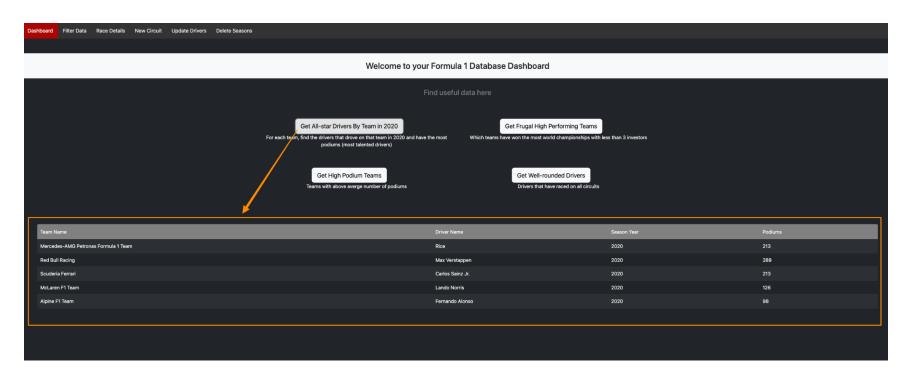
Code Reference(s)

/Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/QueryBuilder.ts::130 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/Server.ts::164

Query Reference

```
static getAggregationWithGroupByQuery(): string {
   let query = `SELECT maxpods.tn AS Team_Name,
                        AS Driver_Name,
      d.name
      dd.season_year
      maxpods.pods AS Podiums
FROM driver d,
      drives_on dsd,
      drove_during dd,
               SELECT dsd.team_name AS tn,
                        max(d.podiums) AS pods
               FROM
                        driver d,
                        drives_on dsd,
                        drove_during dd
               WHERE
                        dd.season_year = 2020
                        dd.driver_id = dsd.driver_id
               AND
               AND
                        d.id = dsd.driver_id
               GROUP BY dsd.team_name) maxpods
WHERE dd.season_year = 2020
AND
      dd.driver_id = dsd.driver_id
      d.id = dsd.driver_id
AND
      d.podiums = maxpods.pods
AND
AND
      dsd.team_name = maxpods.tn`;
   return query;
```

Aggregation with Group By Query



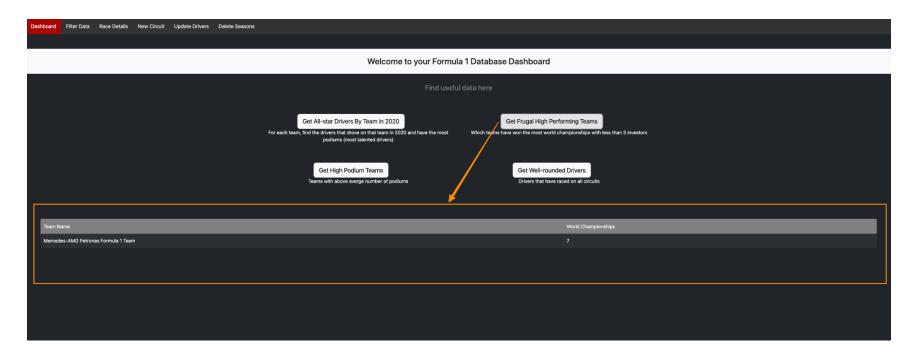
Aggregation with Having DETAILS

Code Reference(s)

/Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/QueryBuilder.ts::157 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/Server.ts::178

Query Reference

Aggregation with Having Query



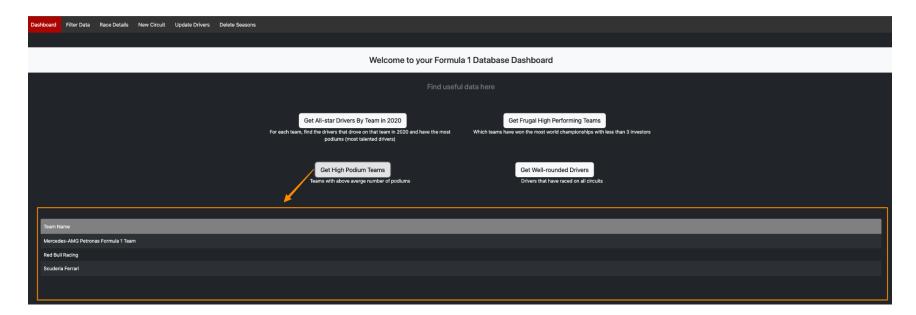
Nested Aggregation with Group By DETAILS

Code Reference(s)

/Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/QueryBuilder.ts::168 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/Server.ts::194

Query Reference

Nested Aggregation with Group By Query



Division DETAILS

Code Reference(s)

/Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/QueryBuilder.ts::180 /Users/kylerich/Developer/304/project_h7w2b_p3v2b_r4h0d/src/server/src/rest/Server.ts::208

Query Reference

```
static getDivisionQuery(): string {
  let query = `SELECT d.name, d.nationality, d.world_championships
              FROM driver d
              WHERE NOT EXISTS ((SELECT name
                                FROM circuit)
                        EXCEPT
                              (SELECT ra.circuit_name
                                 FROM raced r, race ra
                                 WHERE d.id = r.driver_id and r.race_id = ra.id))`
 return query;
```

Division Query

