```
BEGIN;


CREATE TABLE IF NOT EXISTS project_giro.bonus_results
(
    bonus_result_id integer NOT NULL,
    stage_bonus_id integer,
    rider_id integer,
    place integer NOT NULL,
    bonus_points integer,
    bonus_time time with time zone,
    PRIMARY KEY (bonus_result_id)
);

COMMENT ON TABLE project_giro.bonus_results
    IS 'this table stores the results of each competitor, taking into account the place occupied, the number of points awarded and the time bonus, if any';

CREATE TABLE IF NOT EXISTS project_giro.classification_categories
(
    category_id integer NOT NULL,
    classification_id integer,
    category_name character varying(30) NOT NULL,
    points_for_first_place integer,
    points_for_second_place integer,
    points_for_third_place integer,
    points_for_fourth_place integer,
    points_for_fifth_place integer,
    points_for_sixth_place integer,
    points_for_seventh_place integer,
    points_for_eighth_place integer,
    points_for_ninth_place integer,
    points_for_tenth_place integer,
    points_for_eleventh_place integer,
    points_for_twelfth_place integer,
    points_for_thirteenth_place integer,
    points_for_fourteenth_place integer,
    points_for_fifteenth_place integer,
    PRIMARY KEY (category_id)
);

COMMENT ON TABLE project_giro.classification_categories
    IS 'this table stores information about the classification categories and the points allocated for each place according to the category';

CREATE TABLE IF NOT EXISTS project_giro.classifications
(
    classification_id integer NOT NULL,
    classification_name character varying(30) NOT NULL,
    description character varying(150),
    jersey_color character varying(15),
    PRIMARY KEY (classification_id)
```

```
);

COMMENT ON TABLE project_giro.classifications
    IS 'this table is to store all possible classifications and shirt colors for them';

CREATE TABLE IF NOT EXISTS project_giro.countries
(
    country_id integer NOT NULL,
    country_name character varying(50) NOT NULL,
    country_code character varying(3) NOT NULL,
    PRIMARY KEY (country_id)
);

COMMENT ON TABLE project_giro.countries
    IS 'countries from which the riders come';

CREATE TYPE race_status AS ENUM ('active', 'finished');

CREATE TABLE IF NOT EXISTS project_giro.race_classifications
(
    race_classification_id integer NOT NULL,
    classification_id integer,
    race_id integer,
    rider_id integer,
    points integer,
    "time" time with time zone,
    rank integer,
    status_race_classification race_status,
    PRIMARY KEY (race_classification_id)
);

COMMENT ON TABLE project_giro.race_classifications
    IS 'this table stores information and resulst of various classification during the race';

CREATE TABLE IF NOT EXISTS project_giro.races
(
    race_id integer NOT NULL,
    year integer NOT NULL,
    start_date date NOT NULL,
    end_date date NOT NULL,
    number_of_stages integer NOT NULL,
    total_distance numeric(10, 2) NOT NULL,
    start_country_id integer,
    finish_country_id integer,
    start_city_id integer NOT NULL,
    finish_city_id integer NOT NULL,
    description character varying(200),
    PRIMARY KEY (race_id)
);

COMMENT ON TABLE project_giro.races
    IS 'this table stores all the information about the race in each year';
```

```
CREATE TABLE IF NOT EXISTS project_giro.riders
(
    rider_id integer NOT NULL,
    first_name character varying(50) NOT NULL,
    last_name character varying(50) NOT NULL,
    date_of_birth date,
    height numeric(5, 2),
    country_of_origin integer,
    team_id integer,
    speciality_id integer,
    national_champion boolean DEFAULT FALSE,
    PRIMARY KEY (rider_id)
);

COMMENT ON TABLE project_giro.riders
    IS 'table with all the athletes participating in the Giro d''Italia in the years 2000-2024';

CREATE TABLE IF NOT EXISTS project_giro.specialities
(
    speciality_id integer NOT NULL,
    speciality_name character varying(30) NOT NULL,
    PRIMARY KEY (speciality_id)
);

COMMENT ON TABLE project_giro.specialities
    IS 'stores a list of available specialties (e.g., "sprinter", "climber", "time-trialist")';

CREATE TABLE IF NOT EXISTS project_giro.sponsors
(
    sponsor_id integer NOT NULL,
    sponsor_name character varying(50),
    sponsor_country_id integer,
    industry character varying(50),
    PRIMARY KEY (sponsor_id)
);

COMMENT ON TABLE project_giro.sponsors
    IS 'this table is designed to store all the sponsors that were in the Giro d''Italia in the years 2000-2024';

CREATE TABLE IF NOT EXISTS project_giro.sponsorships
(
    sponsorship_id integer NOT NULL,
    classification_id integer,
    sponsor_id integer,
    race_id integer,
    amount numeric(10, 2),
    description character varying(200),
    PRIMARY KEY (sponsorship_id)
);
```

```
COMMENT ON TABLE project_giro.sponsorships
    IS 'the sponsorship table allows you to associate a particular sponsor with a specific classification in a specific race year';

CREATE TABLE IF NOT EXISTS project_giro.stage_bonuses
(
    stage_bonus_id integer NOT NULL,
    stage_id integer,
    category_id integer,
    location character varying(50),
    PRIMARY KEY (stage_bonus_id)
);

COMMENT ON TABLE project_giro.stage_bonuses
    IS 'it contains basic information about each bonus, including the bonus category (e.g., 1st category, 2nd category) and the number of places that are scored';

CREATE TABLE IF NOT EXISTS project_giro.stages
(
    stage_id integer NOT NULL,
    race_id integer,
    stage_number integer NOT NULL,
    stage_date date NOT NULL,
    stage_type_id integer,
    start_city_id integer,
    finish_city_id integer,
    distance numeric(10, 2) NOT NULL,
    elevation_gain numeric(10, 2) NOT NULL,
    stage_time_limit time with time zone,
    PRIMARY KEY (stage_id)
);

COMMENT ON TABLE project_giro.stages
    IS 'this table is intended to store all the stages that have taken place in the Giro d"Italia in the years 2000-2024';


CREATE TABLE IF NOT EXISTS project_giro.stage_results
(
    stage_result_id integer NOT NULL,
    rider_id integer,
    stage_id integer,
    classification_id integer,
    "position" integer NOT NULL,
    "time" time with time zone NOT NULL,
    points integer,
    status stage_status,
    PRIMARY KEY (stage_result_id)
);

COMMENT ON TABLE project_giro.stage_results
    IS 'this table is designed to store the results of each competitor for each stage of the race';
```

```
CREATE TABLE IF NOT EXISTS project_giro.stage_types
(
    stage_type_id integer NOT NULL,
    stage_type_name character varying(50) NOT NULL,
    PRIMARY KEY (stage_type_id)
);

COMMENT ON TABLE project_giro.stage_types
    IS 'this table stores the possible types of stages';

CREATE TABLE IF NOT EXISTS project_giro.start_and_finish_cities
(
    city_id integer NOT NULL,
    start_city character varying(50) NOT NULL,
    finish_city character varying(50) NOT NULL,
    PRIMARY KEY (city_id)
);

COMMENT ON TABLE project_giro.start_and_finish_cities
    IS 'a table storing the cities where the starts and finishes of the stages were held';

CREATE TABLE IF NOT EXISTS project_giro.teams
(
    team_id integer NOT NULL,
    team_name character varying(60) NOT NULL,
    team_country_id integer,
    start_year integer,
    end_year integer,
    PRIMARY KEY (team_id)
);

COMMENT ON TABLE project_giro.teams
    IS 'the teams that took part in the Giro d"Italia in the years 2000-2024';

CREATE TABLE IF NOT EXISTS project_giro.team_riders
(
    team_rider_id integer NOT NULL,
    team_id integer,
    rider_id integer,
    race_id integer,
    role_team character varying(30) NOT NULL,
    PRIMARY KEY (team_rider_id)
);

COMMENT ON TABLE project_giro.team_riders
    IS 'a table linking players to teams in different year (since players can change teams)';

ALTER TABLE IF EXISTS project_giro.bonus_results
    ADD CONSTRAINT stage_bonus_id FOREIGN KEY (stage_bonus_id)
    REFERENCES project_giro.stage_bonuses (stage_bonus_id) MATCH SIMPLE
    ON UPDATE NO ACTION
```

```
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.bonus_results
    ADD CONSTRAINT rider_id FOREIGN KEY (rider_id)
    REFERENCES project_giro.riders (rider_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.classification_categories
    ADD CONSTRAINT classification_id_cc FOREIGN KEY (classification_id)
    REFERENCES project_giro.classifications (classification_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.race_classifications
    ADD CONSTRAINT classification_id_rc FOREIGN KEY (classification_id)
    REFERENCES project_giro.classifications (classification_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.race_classifications
    ADD CONSTRAINT race_id_rc FOREIGN KEY (race_id)
    REFERENCES project_giro.races (race_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.race_classifications
    ADD CONSTRAINT rider_id_rc FOREIGN KEY (rider_id)
    REFERENCES project_giro.riders (rider_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.races
    ADD CONSTRAINT start_country_id FOREIGN KEY (start_country_id)
    REFERENCES project_giro.countries (country_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;
```

```
ALTER TABLE IF EXISTS project_giro.races
    ADD CONSTRAINT finish_country_id FOREIGN KEY (finish_country_id)
    REFERENCES project_giro.countries (country_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.races
    ADD CONSTRAINT start_city_id FOREIGN KEY (start_city_id)
    REFERENCES project_giro.start_and_finish_cities (city_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.races
    ADD CONSTRAINT finish_city_id FOREIGN KEY (finish_city_id)
    REFERENCES project_giro.start_and_finish_cities (city_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.riders
    ADD CONSTRAINT country_of_origin FOREIGN KEY (country_of_origin)
    REFERENCES project_giro.countries (country_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.riders
    ADD CONSTRAINT team_id FOREIGN KEY (team_id)
    REFERENCES project_giro.teams (team_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.riders
    ADD CONSTRAINT speciality_id FOREIGN KEY (speciality_id)
    REFERENCES project_giro.specialities (speciality_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.sponsors
    ADD CONSTRAINT sponsor_country_id FOREIGN KEY (sponsor_country_id)
    REFERENCES project_giro.countries (country_id) MATCH SIMPLE
    ON UPDATE NO ACTION
```

```
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.sponsorships
    ADD CONSTRAINT classification_id_sponsor FOREIGN KEY (classification_id)
    REFERENCES project_giro.classifications (classification_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.sponsorships
    ADD CONSTRAINT sponsor_id FOREIGN KEY (sponsor_id)
    REFERENCES project_giro.sponsors (sponsor_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.sponsorships
    ADD CONSTRAINT race_id_sponsor FOREIGN KEY (race_id)
    REFERENCES project_giro.races (race_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.stage_bonuses
    ADD CONSTRAINT stage_id_sb FOREIGN KEY (stage_id)
    REFERENCES project_giro.stages (stage_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.stage_bonuses
    ADD CONSTRAINT category_id_sb FOREIGN KEY (category_id)
    REFERENCES project_giro.classification_categories (category_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.stages
    ADD CONSTRAINT race_id_stage FOREIGN KEY (race_id)
    REFERENCES project_giro.races (race_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;
```

```
ALTER TABLE IF EXISTS project_giro.stages
    ADD CONSTRAINT stage_type_id FOREIGN KEY (stage_type_id)
    REFERENCES project_giro.stage_types (stage_type_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.stages
    ADD CONSTRAINT start_city_id_stage FOREIGN KEY (start_city_id)
    REFERENCES project_giro.start_and_finish_cities (city_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.stages
    ADD CONSTRAINT finish_city_id_stage FOREIGN KEY (finish_city_id)
    REFERENCES project_giro.start_and_finish_cities (city_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.stage_results
    ADD CONSTRAINT rider_id_sr FOREIGN KEY (rider_id)
    REFERENCES project_giro.riders (rider_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.stage_results
    ADD CONSTRAINT stage_id_sr FOREIGN KEY (stage_id)
    REFERENCES project_giro.stages (stage_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.stage_results
    ADD CONSTRAINT classification_id_sr FOREIGN KEY (classification_id)
    REFERENCES project_giro.classifications (classification_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.teams
    ADD CONSTRAINT team_country_id FOREIGN KEY (team_country_id)
    REFERENCES project_giro.countries (country_id) MATCH SIMPLE
    ON UPDATE NO ACTION
```

```
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.team_riders
    ADD CONSTRAINT team_id_tr FOREIGN KEY (team_id)
    REFERENCES project_giro.teams (team_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.team_riders
    ADD CONSTRAINT rider_id_tr FOREIGN KEY (rider_id)
    REFERENCES project_giro.riders (rider_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS project_giro.team_riders
    ADD CONSTRAINT race_id_tr FOREIGN KEY (race_id)
    REFERENCES project_giro.races (race_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;

END;
```