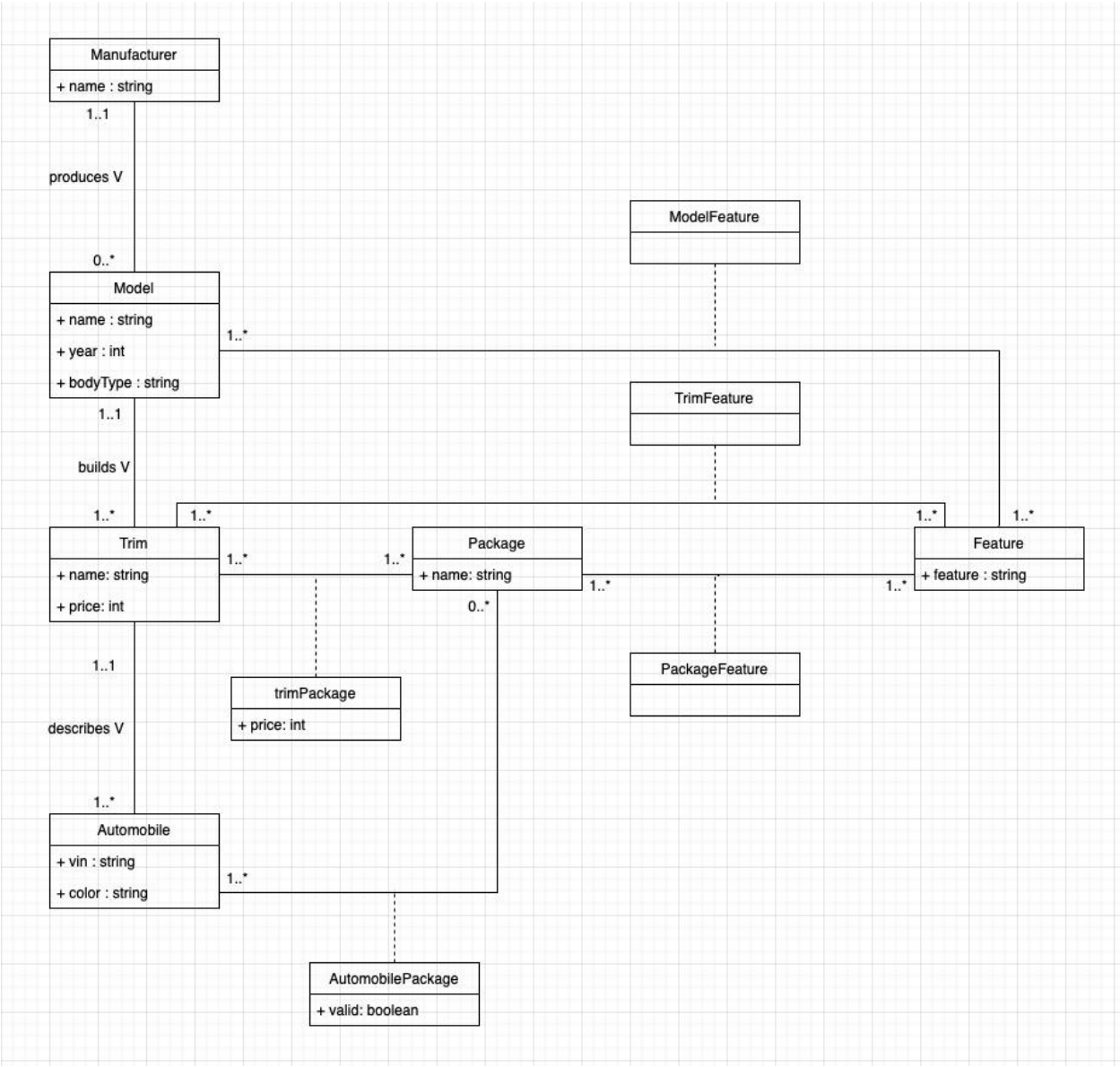
CECS 323-02

Project 2 - Drive My Car

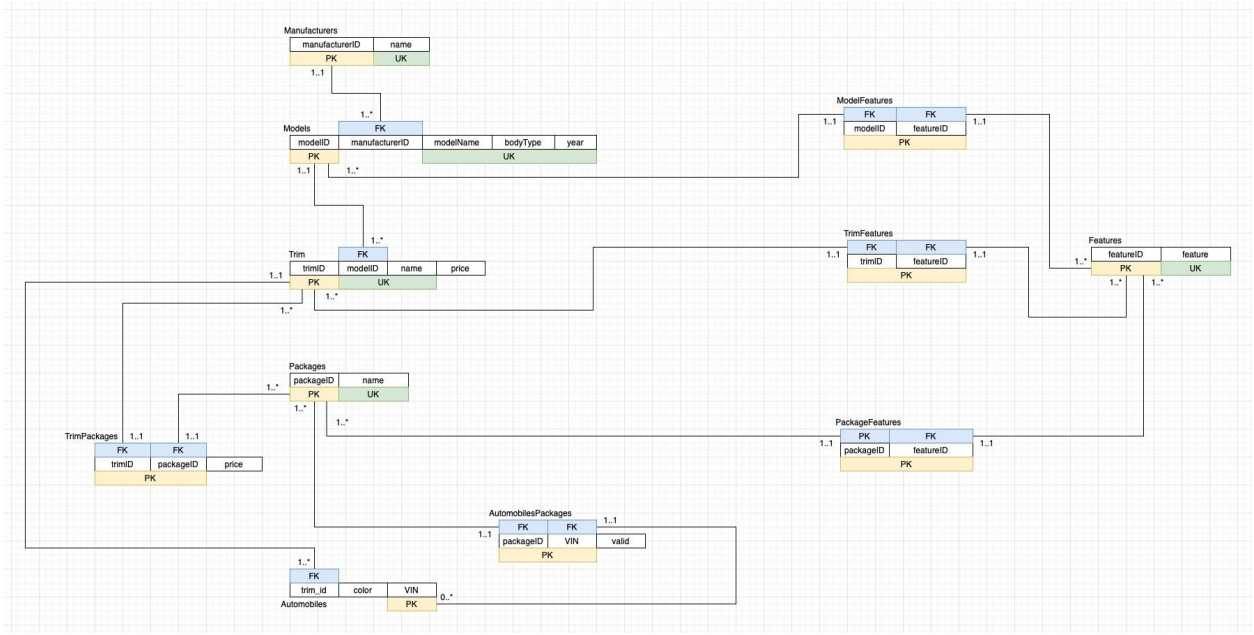
Michael Ching, Javier Sanchez

Due: 4/9/2022

UML:


Manufacturer
+ name : string
1..1
produces V
0..*
Model
+ name : string
+ year : int
+ bodyType : string
1..1
builds V
1..*
ModelFeature
TrimFeature
1..*
1..*
1..*
1..*
Trim
+ name: string
+ price: int
1..*
1..*
Package
+ name: string
1..*
1..*
Feature
+ feature : string
0..*
1..1
describes V
1..*
trimPackage
+ price: int
PackageFeature
Automobile
+ vin : string
+ color : string
1..*
AutomobilePackage
+ valid: boolean

Relation Scheme:



DDL Commands:

```
create table packages
(
    "packageID" serial
        constraint packages_pk
            primary key,
    name        varchar(100) not null
);

alter table packages
    owner to postgres;

create unique index packages_name_uindex
    on packages (name);

create table manufacturers
(
```

```
    "manufacturerID" serial
        constraint manufacturers_pk
            primary key,
    name            varchar(100) not null
);

alter table manufacturers
    owner to postgres;

create unique index manufacturers_name_uindex
    on manufacturers (name);

create table models
(
    "modelID"       serial
        constraint models_pk
            primary key,
```

```
    "manufacturerID" integer      not null
        constraint models___fk1
            references manufacturers,
    "modelName"      varchar(100) not null,
    "bodyType"       varchar(100) not null,
    year             integer      not null
);


alter table models
    owner to postgres;


create table trims
(
    "trimID"  integer      not null
        constraint trims_pk
            primary key,
    "modelID" integer      not null
```

```
        constraint trims___fk1
            references models,
    name      varchar(100) not null,
    price     integer      not null
);


alter table trims
    owner to postgres;


create unique index "trims__PK"
    on trims ("modelID", name);


create table automobiles
(
    "trimID" integer      not null
        constraint automobiles___fk1
            references trims,
```

```
    color     varchar(100),
    vin       varchar(100) not null
        constraint automobiles_pk
            primary key
);


alter table automobiles
    owner to postgres;


create table "trimPackages"
(
    "packageID" integer not null
        constraint trimpackages_packages_packageid_fk
            references packages,
    "trimID"    integer not null
        constraint trimpackages_trims_trimid_fk
            references trims,
```

```
    price       integer,
    constraint trimpackages_pk
        unique ("packageID", "trimID")
);

alter table "trimPackages"
    owner to postgres;

create table features
(
    feature     varchar(100) not null,
    "featureID" serial
        constraint features_pk
            primary key
);

alter table features
```

```
    owner to postgres;



create table "modelFeatures"

(

    "featureID " integer not null

        constraint modelfeatures___fk2

            references features,

    "modelID"    integer not null

        constraint modelfeature_models_modelid_fk

            references models,

    constraint modelfeature_pk

        primary key ("featureID ", "modelID")

);



alter table "modelFeatures"

    owner to postgres;
```

```
create table "trimFeatures"
(
    "trimID"    integer not null
        constraint trimfeature_trims_trimid_fk
            references trims,
    "featureID" integer not null
        constraint trimfeatures_pk
            primary key
        constraint trimfeatures___fk2
            references features,
    constraint trimfeature_pk
        unique ("trimID", "featureID")
);

alter table "trimFeatures"
    owner to postgres;
```

```
create table "packageFeatures"

(

    "packageID" integer not null

        constraint packagefeature_packages_packageid_fk

            references packages,

    "featureID" integer not null

        constraint packagefeatures___fk2

            references features,

    constraint packagefeature_pk

        primary key ("packageID", "featureID")

);



alter table "packageFeatures"

    owner to postgres;



create table "automobilePackages"

(
```

```
    "packageID" integer      not null
        constraint automobilepackages_packages_packageid_fk
            references packages,
    vin         varchar(100) not null
        constraint automobilepackages___fk2
            references automobiles,
    valid       boolean      not null
);


alter table "automobilePackages"
    owner to postgres;


create function check_automobile_package() returns trigger
    language plpgsql
as
$$
BEGIN
```

```
  IF ( new."packageID" not in

      (select tP."packageID"

       from "trimPackages" as tP

       where "trimID" =

        (select distinct automobiles."trimID"

         from automobiles

         inner join project2."automobilePackages" aP on automobiles.vin = NEW.vin) ) )

        THEN

     RAISE EXCEPTION 'Invalid package with trim of automobile.';

  END IF;



  RETURN NEW;

END;

$$;



alter function check_automobile_package() owner to postgres;
```

```
create trigger automobilepackage_insert_select
    before insert
    on "automobilePackages"
    for each row
execute procedure check_automobile_package();
```

SQL Select Statements:

```
--1

select m2.name as ManufacturerName, m."modelName", m.year, t.name, vin

from automobiles

inner join trims t on t."trimID" = automobiles."trimID"

inner join models m on m."modelID" = t."modelID"

inner join manufacturers m2 on m2."manufacturerID" = m."manufacturerID";
```

```
--2

select "modelName", year, min(price)

from trims

inner join models m on m."modelID" = trims."modelID"

inner join manufacturers m2 on m2."manufacturerID" = m."manufacturerID"

where m2.name = 'Toyota'

group by "modelName", m.year;
```

```
--3
```

```sql
select count(*)
from automobiles
where automobiles."trimID" NOT IN
(select "trimID"
from "trimFeatures"
where "featureID" IN
(select "featureID"
from features
where feature LIKE '%leather%')
group by "trimID");
```

```sql
--4
select vin
from automobiles
inner join trims t on t."trimID" = automobiles."trimID"
inner join "trimPackages" tP on t."trimID" = tP."trimID"
where t.price + tP.price =
```

```
(select max(t.price + tP.price)

from automobiles

inner join trims t on t."trimID" = automobiles."trimID"

inner join "trimPackages" tP on t."trimID" = tP."trimID");
```