create table if not exists public.sys\_user

(

id bigint not null,

password varchar(100) not null,

version int not null,

create\_time timestamp not null,

update\_time timestamp not null,

name varchar(20) not null,

role varchar(20) not null,

email varchar(50) not null,

constraint sys\_user\_pkey primary key (id),

constraint sys\_user\_id\_name\_key unique (id)

include(name)

)

create table if not exists public.artist

(

id bigint not null,

name varchar(20) not null,

version int not null,

create\_time timestamp not null,

update\_time timestamp not null,

constraint artist\_pkey primary key (id),

constraint artist\_id\_name\_key unique (id)

include(name)

)

create table if not exists public.album

(

id bigint not null,

name varchar(50) not null,

version int not null,

create\_time timestamp not null,

update\_time timestamp not null,

constraint album\_pkey primary key (id),

constraint album\_id\_name\_key unique (id)

include(name)

)

create table if not exists public.track

(

id bigint not null,

name varchar(50) not null,

time\_length varchar(10) not null,

version int not null,

create\_time timestamp not null,

update\_time timestamp not null,

checked boolean NOT NULL DEFAULT false,

file uuid,

constraint track\_pkey primary key (id),

constraint track\_id\_name\_key unique (id)

include(name)

)

create table if not exists public.make

(

id bigint not null,

album\_id bigint not null,

artist\_ id bigint not null,

version int not null,

year varchar(4) not null,

create\_time timestamp not null,

update\_time timestamp not null,

constraint make\_pkey primary key (id),

constraint make\_id\_album\_id\_artist\_id\_key unique (id)

include(album\_id, artist\_id),

constraint make\_album\_id\_fkey foreign key (album\_id)

references public.album (id) match simple

on update cascade

on delete cascade

not valid,

constraint make\_artist\_id\_fkey foreign key (artist\_id)

references public.artist (id) match simple

on update cascade

on delete cascade

not valid

)

create table if not exists public.belong\_to

(

id bigint not null,

track\_ id bigint not null,

album\_ id bigint not null,

version int not null,

create\_time timestamp not null,

update\_time timestamp not null,

constraint belong\_to\_pkey primary key (id),

constraint belong\_to\_id\_track\_id\_album\_id\_key unique (id)

include(track\_id, album\_id),

constraint belong\_to\_album\_id\_fkey foreign key (album\_id)

references public.album (id) match simple

on update cascade

on delete cascade

not valid,

constraint belong\_to\_track\_id\_fkey foreign key (track\_id)

references public.track (id) match simple

on update cascade

on delete cascade

not valid

)

create table if not exists public.collect

(

id bigint not null,

user\_ id bigint not null,

album\_ id bigint not null,

version int not null,

collected\_time varchar(30) not null,

create\_time timestamp not null,

update\_time timestamp not null,

constraint collect\_pkey primary key (id),

constraint collect\_id\_user\_id\_album\_id\_key unique (id)

include(user\_id, album\_id),

constraint collect\_album\_id\_fkey foreign key (album\_id)

references public.album (id) match simple

on update cascade

on delete cascade

not valid,

constraint collect\_user\_id\_fkey foreign key (user\_id)

references public.sys\_user (id) match simple

on update cascade

on delete cascade

not valid

)

create table if not exists public.comment

(

id bigint not null,

user\_ id bigint not null,

album\_ id bigint not null,

version int not null,

context varchar(300) not null,

commented\_time varchar(30) not null,

create\_time timestamp not null,

update\_time timestamp not null,

parent\_id bigint,

root\_parent\_id bigint,

constraint comment\_pkey primary key (id),

constraint comment\_id\_user\_id\_album\_id\_key unique (id)

include(user\_id, album\_id),

constraint comment\_album\_id\_fkey foreign key (album\_id)

references public.album (id) match simple

on update cascade

on delete cascade

not valid,

constraint comment\_user\_id\_fkey foreign key (user\_id)

references public.sys\_user (id) match simple

on update cascade

on delete cascade

not valid

)

create table if not exists public.open

(

id bigint not null,

user\_ id bigint not null,

album\_ id bigint not null,

version int not null,

index\_last\_played\_track integer not null,

create\_time timestamp not null,

update\_time timestamp not null,

constraint open\_pkey primary key (id),

constraint open\_id\_user\_id\_album\_id\_key unique (id)

include(user\_id, album\_id),

constraint open\_album\_id\_fkey foreign key (album\_id)

references public.album (id) match simple

on update cascade

on delete cascade

not valid,

constraint open\_user\_id\_fkey foreign key (user\_id)

references public.sys\_user (id) match simple

on update cascade

on delete cascade

not valid

)

create table if not exists public.listen

(

id bigint not null,

user\_id bigint not null,

track\_ id bigint not null,

version int not null,

latest\_date\_time\_playback\_began varchar(30) not null,

create\_time timestamp not null,

update\_time timestamp not null,

constraint listen\_pkey primary key (id),

constraint listen\_id\_user\_id\_track\_id\_key unique (id)

include(user\_id, track\_id),

constraint listen\_track\_id\_fkey foreign key (track\_id)

references public.track (id) match simple

on update cascade

on delete cascade

not valid,

constraint listen\_user\_id\_fkey foreign key (user\_id)

references public.sys\_user (id) match simple

on update cascade

on delete cascade

not valid

)