

# Final Report

Faculty Sponsor:

Suneuy Kim - San Jose State University Professor  
San Jose State University - College of Engineering

Principal Investigators:

Diego Cruz  
Jared Park  
Andrew Tedijanto  
Shuti Wang

**Team CPTW**

**3. Database schema in the form of SQL CREATE TABLE. Specify key constraints and foreign key constraints in the schema.**

```
DROP DATABASE IF EXISTS PLUGINS;
```

```
CREATE DATABASE PLUGINS;
```

```
USE PLUGINS;
```

```
DROP TABLE IF EXISTS Plugins;
```

```
CREATE TABLE Plugins(
```

```
plugin VARCHAR(50), # plugin uses as join attributes(name) in generator and fx table
```

```
developer VARCHAR(30),
```

```
type VARCHAR(20), # fx or generator
```

```
PRIMARY KEY(plugin)
```

```
);
```

```
DROP TABLE IF EXISTS Generator;
```

```
CREATE TABLE Generator(
```

```
name VARCHAR(50) PRIMARY KEY, # name of the plugin
```

```
format VARCHAR(10),
```

```
subtype VARCHAR(50),
```

```
isFree BOOLEAN DEFAULT FALSE,
```

```
hasLicense BOOLEAN DEFAULT FALSE,
```

```
lastUpdate timestamp not null on update current_timestamp # 0000-00-00 00:00:00
```

```
);
```

```
DROP TABLE IF EXISTS Fx;
```

```
CREATE TABLE Fx(
```

```
name VARCHAR(50) PRIMARY KEY, # name of the plugin
```

```
format VARCHAR(10),
```

```
subtype VARCHAR(50),
```

```
isFree BOOLEAN DEFAULT FALSE,
```

```
hasLicense BOOLEAN DEFAULT FALSE,
```

```
lastUpdate timestamp not null on update current_timestamp
```

```
);
```

```
DROP TABLE IF EXISTS User;
```

```
CREATE TABLE User(
```

```
uID INT AUTO_INCREMENT,
```

```
name VARCHAR(30),
age INT,
downLoaded INT,
PRIMARY KEY(uID)
);
```

```
DROP TABLE IF EXISTS Downloaded;
CREATE TABLE Downloaded(
uID INT,
plugin VARCHAR(50),
downloadDate timestamp not null on update current_timestamp,
PRIMARY KEY(uID, plugin, downloadDate),
FOREIGN KEY(uID) references User (uID) on delete cascade,
FOREIGN KEY(plugin) references Plugins (plugin) on delete cascade
);
```

```
DROP TABLE IF EXISTS Archive;
CREATE TABLE Archive(
name VARCHAR(50) primary key, # name of the plugin
format VARCHAR(10),
developer VARCHAR(30),
type VARCHAR(20), # fx or generator
subtype VARCHAR(50),
isFree BOOLEAN DEFAULT FALSE,
hasLicense BOOLEAN DEFAULT FALSE,
lastUpdate timestamp not null on update current_timestamp,
uID INT,
downloadDate timestamp on update current_timestamp
);
```

**4. Indicate if your team used a public data set or not. If a public data set is used, specify its URL.**

Our project is a representation of plugin management for audio processing software commonly used in audio engineering applications and music production. The data set is inspired by the current software inside of the King Library 4th Floor Sound Studio managed by Student Computing Services as well as the software inside of the personal workstation of Diego Cruz. The data set structure is partially inspired by the plugin databases that exist for Digital Audio Workstation softwares such as Image-Line FL Studio and Cockos REAPER. The group sought to improve upon these already-established plugin databases by introducing new attributes as well as introducing a modified structure expressed through tables.

## 5. Descriptions to show relation schemas are in BCNF or in 3NF.

Step 1: start with all attributes

$R(\text{plugin}, \text{format}, \text{developer}, \text{type}, \text{subtype}, \text{isFree}, \text{hasLicense}, \text{lastUpdate}, \text{uID}, \text{name}, \text{age}, \text{downloaded}, \text{downloadDate},)$

$\text{plugin} \rightarrow \text{developer}, \text{type}$

$\text{plugin}(\text{name}) \rightarrow \text{format}, \text{subtype}, \text{isFree}, \text{hasLicense}, \text{lastUpdate}$

$\text{uID} \rightarrow \text{name}, \text{age}, \text{downloaded}$

$\text{uID}, \text{plugin}, \text{downloadDate} \rightarrow \text{uID}, \text{plugin}, \text{downloadDate}$

$\{\text{plugin}\}^+ : \{\text{plugin}, \text{developer}, \text{type}, \text{format}, \text{subtype}, \text{isFree}, \text{hasLicense}, \text{lastUpdate}\}$  is not a super key, BCNF violation.

$\{\text{uID}\}^+ : \{\text{uID}, \text{name}, \text{age}, \text{downloaded}\}$  is not a super key, BCNF violation

$\{\text{uID}, \text{plugins}, \text{downloadDate}\}^+ : \{\text{uID}, \text{name}, \text{age}, \text{downloaded}, \text{plugins}, \text{downloadDate}\}$  is not a super key, BCNF violation.

Step 2: decompose

Decompose 1:

$R(\text{plugin}, \text{developer}, \text{type}, \text{format}, \text{subtype}, \text{isFree}, \text{hasLicense}, \text{lastUpdate})$

$\text{plugin} \rightarrow \text{developer}, \text{type}$

$\text{plugin}(\text{name}) \rightarrow \text{format}, \text{subtype}, \text{isFree}, \text{hasLicense}, \text{lastUpdate}$

$\{\text{plugin}\}^+ : \{\text{plugin}, \text{developer}, \text{type}, \text{format}, \text{subtype}, \text{isFree}, \text{hasLicense}, \text{lastUpdate}\}$  is a super key, no BCNF violation.

Since we have 2 types, we decided to further decompose into 2 tables based on the type (Fx, Generator)

Each type will have its attributes: ( $\text{plugin}(\text{name}) \rightarrow \text{format}, \text{subtype}, \text{isFree}, \text{hasLicense}, \text{lastUpdate}$ )

Decompose 2:

$R(\text{uID}, \text{name}, \text{age}, \text{downloaded}, \text{plugin}, \text{downloadDate}, \text{downloadDate})$

$\text{uID} \rightarrow \text{name, age, downloaded}$

$\text{uID, plugin, downloadDate} \rightarrow \text{uID, plugin, downloadDate}$

$\{\text{uID}\}^+ : \{\text{uID, name, age, downloaded}\}$  is not a super key, BCNF violation

$\{\text{uID, plugins, downloadDate}\}^+ : \{\text{uID, plugins, downloadDate, name, age, downloaded}\}$  is super key, no BCNF violation.

Decompose 3

$R(\text{uID, name, age, downloaded})$

$\text{uID} \rightarrow \text{name, age, downloaded}$

$\{\text{uID}\}^+ : \{\text{uID, name, age, downloaded}\}$  is a super key, no BCNF violation

Step 3 all the tables we have:

Plugins(plugin, developer, type)

Generator(name, format, subtype, isFree, hasLicense, lastUpdate)

Fx(name, format, subtype, isFree, hasLicense, lastUpdate)

User(uID, name, age, downloaded)

Downloaded(uID, plugin, downloadDate)

Archive table will have attributes :Archive(plugin(name), format, developer, type, subtype, isFree, hasLicense, lastUpdate, uID, downloadDate) with plugin(name) as the primary key.

## 6. Screenshots of all relations after populating initial data. Label each relation clearly.

```
mysql> select * from generator;
+-----+-----+-----+-----+-----+-----+-----+
| name | format | subtype | isFree | hasLicense | lastUpdate |
+-----+-----+-----+-----+-----+-----+
| Ample Bass P Lite 2 | VST2 | ROMpler | 1 | 0 | 2019-07-22 00:00:00 |
| Ample Guitar M Lite | VST2 | ROMpler | 1 | 0 | 2018-04-04 00:00:00 |
| Delay Lama | VST2 | Softsynth | 1 | 0 | 2021-05-12 00:00:00 |
| Dexed | VST2 | Softsynth | 1 | 0 | 2019-07-22 00:00:00 |
| Keyzone Classic | VST2 | ROMpler | 1 | 0 | 2020-01-25 00:00:00 |
| LABS | VST3 | ROMpler | 1 | 0 | 2017-06-19 00:00:00 |
| LABS Piano | VST3 | ROMpler | 1 | 0 | 2017-06-19 00:00:00 |
| Papu | VST2 | Softsynth | 1 | 0 | 2020-01-25 00:00:00 |
| PhasePlant | VST3 | Softsynth | 0 | 1 | 2020-11-28 00:00:00 |
| Podolski | VST2 | Softsynth | 1 | 0 | 2018-12-01 00:00:00 |
| Serum | VST2 | Softsynth | 0 | 1 | 2017-11-29 00:00:00 |
| Sforzando | VST2 | ROMpler | 1 | 0 | 2020-05-13 00:00:00 |
| Synth1 | VST2 | Softsynth | 1 | 0 | 2017-09-22 00:00:00 |
| TAL-BassLine | VST2 | Softsynth | 1 | 0 | 2017-02-08 00:00:00 |
| TyrellN6 | VST2 | Softsynth | 1 | 0 | 2018-12-01 00:00:00 |
| VCV Rack | AAX | Softsynth | 1 | 0 | 2021-05-12 00:00:00 |
| Vital | VST3 | Softsynth | 1 | 0 | 2020-11-24 00:00:00 |
| Zebralette | VST3 | Softsynth | 1 | 0 | 2018-12-01 00:00:00 |
+-----+-----+-----+-----+-----+-----+
18 rows in set (0.01 sec)
```

```
MySQL [localhost:3306 ssl] plugins SQL > select * from fx;
+-----+-----+-----+-----+-----+-----+
| name | format | subtype | isFree | hasLicense | lastUpdate |
+-----+-----+-----+-----+-----+-----+
| 3-Band EQ | VST3 | EQ | 1 | 0 | 2020-11-28 00:00:00 |
| 8 Bit Shaper | VST2 | Distortion | 1 | 0 | 2017-11-29 00:00:00 |
| ADHD Leveling Tool | VST2 | Dynamics | 1 | 0 | 2018-12-01 00:00:00 |
| Bitcrush | VST3 | Distortion | 0 | 1 | 2020-11-28 00:00:00 |
| Bitter Sweet V3 | VST3 | Dynamics | 1 | 0 | 2019-04-11 00:00:00 |
| BX_Cleansweep V2 | VST3 | Filter | 1 | 1 | 2017-05-24 00:00:00 |
| BX_Rockrack V3 Player | VST3 | Distortion | 1 | 1 | 2017-05-24 00:00:00 |
| CamelCrusher | VST2 | Distortion | 1 | 0 | 2018-12-01 00:00:00 |
| CarveEQ | VST3 | EQ | 0 | 1 | 2020-11-28 00:00:00 |
| Chorus | VST3 | Chorus | 1 | 0 | 2020-11-28 00:00:00 |
| Comb Filter | VST3 | Filter | 0 | 1 | 2020-11-28 00:00:00 |
| Compressor | VST3 | Dynamics | 0 | 1 | 2020-11-28 00:00:00 |
| Convolver | VST3 | Reverb | 0 | 1 | 2020-11-28 00:00:00 |
| Correlometer | VST2 | Monitoring | 1 | 0 | 2017-09-22 00:00:00 |
| Delay | VST3 | Delay | 1 | 0 | 2020-11-28 00:00:00 |
| Delta Modulator | VST2 | Distortion | 1 | 0 | 2017-11-29 00:00:00 |
| Disperser | VST3 | Filter | 0 | 1 | 2020-11-28 00:00:00 |
| Distortion | VST3 | Distortion | 0 | 1 | 2020-11-28 00:00:00 |
| Dynamics | VST3 | Dynamics | 0 | 1 | 2020-11-28 00:00:00 |
| Ensemble | VST3 | Chorus | 0 | 1 | 2020-11-28 00:00:00 |
| Faturator | VST3 | Distortion | 0 | 1 | 2020-11-28 00:00:00 |
| Filter | VST3 | Filter | 0 | 1 | 2020-11-28 00:00:00 |
| Flanger | VST3 | Filter | 0 | 1 | 2020-11-28 00:00:00 |
| Formant Filter | VST3 | Filter | 0 | 1 | 2020-11-28 00:00:00 |
| Fresh Air | VST3 | Misc | 1 | 0 | 2020-11-27 00:00:00 |
| Gain | VST3 | Utility | 1 | 0 | 2020-11-28 00:00:00 |
| Gate | VST3 | Gate | 0 | 1 | 2020-11-28 00:00:00 |
| GClip | VST2 | Distortion | 1 | 0 | 2020-12-27 00:00:00 |
| Glitch | VST2 | Host | 1 | 0 | 2019-04-11 00:00:00 |
| GSnap | VST2 | Pitch Correction | 1 | 0 | 2018-01-17 00:00:00 |
| Haas | VST3 | Imaging | 0 | 1 | 2020-11-28 00:00:00 |
| Krush | VST3 | Distortion | 1 | 0 | 2017-06-19 00:00:00 |
| Ladder Filter | VST3 | Filter | 0 | 1 | 2020-11-28 00:00:00 |
| Limiter | VST3 | Dynamics | 1 | 0 | 2020-11-28 00:00:00 |
| Loudness Meter II | VST3 | Monitoring | 1 | 0 | 2017-09-22 00:00:00 |
| Loudness Meter II Pro | VST3 | Monitoring | 0 | 1 | 2018-09-20 00:00:00 |
| MAutoPan | VST3 | Modulation | 1 | 0 | 2017-04-05 00:00:00 |
| MAutoPitch | VST3 | Pitch Correction | 1 | 0 | 2017-04-05 00:00:00 |
| MBandPass | VST3 | Filter | 1 | 0 | 2017-04-05 00:00:00 |
+-----+-----+-----+-----+-----+-----+
```

MBitFun	VST3	Distortion	1	0	2017-04-05 00:00:00
MComb	VST3	Filter	1	0	2017-04-05 00:00:00
MCompressor	VST3	Dynamics	1	0	2017-04-05 00:00:00
MConvolutionEZ	VST3	Reverb	1	0	2017-04-05 00:00:00
Melodyne Essential	VST3	Pitch Correction	0	1	2020-08-21 00:00:00
MEqualizer	VST3	EQ	1	0	2017-04-05 00:00:00
MFlanger	VST3	Filter	1	0	2017-04-05 00:00:00
MFreqShifter	VST3	Misc	1	0	2017-04-05 00:00:00
MPhaser	VST3	Filter	1	0	2017-04-05 00:00:00
MRingModulator	VST3	Modulation	1	0	2017-04-05 00:00:00
MSaturator	VST3	Distortion	1	0	2017-04-05 00:00:00
MTremolo	VST3	Modulation	1	0	2017-04-05 00:00:00
Multipass	VST3	Host	0	1	2020-11-28 00:00:00
MVibrato	VST3	Pitch Shift	1	0	2017-04-05 00:00:00
MWaveFolder	VST3	Distortion	1	0	2017-04-05 00:00:00
Niveau Filter	VST3	Filter	1	1	2017-05-24 00:00:00
Nonlinear Filter	VST3	Filter	0	1	2020-11-28 00:00:00
OldSkoolVerb	VST2	Reverb	0	0	2017-09-22 00:00:00
OTT	VST2	Dynamics	1	0	2017-11-29 00:00:00
Ozone 9 Advanced	VST3	Mastering	0	1	2020-08-21 00:00:00
Ozone Imager II	VST3	Imaging	1	1	2017-06-19 00:00:00
PanCake II	VST2	Modulation	1	0	2020-12-27 00:00:00
Phase Distortion	VST3	Distortion	0	1	2020-11-28 00:00:00
Phaser	VST3	Filter	0	1	2020-11-28 00:00:00
Pitch Shifter	VST3	Pitch Shift	0	1	2020-11-21 00:00:00
Pro-C 2	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-DS	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-G	VST3	Gate	0	1	2021-08-08 00:00:00
Pro-L 2	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-MB	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-Q 3	VST3	EQ	0	1	2019-04-30 00:00:00
Pro-R	VST3	Reverb	0	1	2020-11-25 00:00:00
Resonator	VST3	Misc	0	1	2020-11-28 00:00:00
Reverb	VST3	Reverb	0	1	2020-11-28 00:00:00
Reverser	VST3	Misc	0	1	2020-11-28 00:00:00
Ring Mod	VST3	Modulation	0	1	2020-11-28 00:00:00
Rough Rider 3	VST2	Dynamics	1	0	2018-10-08 00:00:00
Saturation Knob	VST3	Distortion	1	1	2017-06-19 00:00:00
Saturn 2	VST3	Distortion	0	1	2021-08-08 00:00:00
Slice EQ	VST3	EQ	0	1	2020-11-28 00:00:00
SnapHeap	VST3	Host	0	0	2020-11-28 00:00:00
SPAN	VST3	Monitoring	1	0	2017-09-22 00:00:00
Stereo	VST3	Imaging	1	0	2020-11-28 00:00:00

TAL Reverb 4	VST2	Reverb	1	0	2020-03-19 00:00:00
TAL-Vocoder	VST2	Misc	1	0	2017-04-05 00:00:00
Tape Stop	VST3	Misc	0	1	2020-11-28 00:00:00
TapeDelay	VST2	Delay	1	0	2018-09-03 00:00:00
TDR Kotelnikov	VST3	Dynamics	1	0	2020-10-16 00:00:00
TDR Molotok	VST3	Dynamics	1	0	2020-10-16 00:00:00
TDR NOVA	VST3	EQ	1	0	2020-10-16 00:00:00
TDR VOS SlickEQ	VST3	EQ	1	0	2020-10-16 00:00:00
Temper	VST2	Dynamics	1	0	2019-04-11 00:00:00
Timeless 3	VST3	Delay	0	1	2021-08-08 00:00:00
Trance Gate	VST3	Gate	0	1	2020-11-28 00:00:00
Transient Shaper	VST3	Dynamics	0	1	2020-11-28 00:00:00
Valhalla FreqEcho	VST3	Delay	1	0	2020-07-11 00:00:00
Valhalla SpaceModulator	VST3	Filter	1	0	2020-07-11 00:00:00
Valhalla SuperMassive	VST3	Reverb	1	0	2020-07-11 00:00:00
Valhalla UberMod	VST3	Chorus	0	1	2020-07-11 00:00:00
Valhalla VintageVerb	VST3	Reverb	0	1	2020-07-11 00:00:00
ValhallaDelay	VST3	Delay	0	1	2020-07-11 00:00:00
ValhallaPlate	VST3	Reverb	0	1	2020-07-11 00:00:00
ValhallaRoom	VST3	Reverb	0	1	2020-07-11 00:00:00
ValhallaShimmer	VST3	Reverb	0	1	2020-07-11 00:00:00
Vinyl	VST3	Distortion	1	1	2017-06-19 00:00:00
Vladg Limiter no.6	VST2	Dynamics	1	0	2020-10-16 00:00:00
Vocal Doubler	VST3	Misc	1	1	2017-06-19 00:00:00
VOLOCO	VST3	Pitch Correction	1	0	2019-07-14 00:00:00

107 rows in set (0.0220 sec)

MySQL localhost:3306 ssl SQL >

```
mysql> select * from Downloaded;
+----+-----+-----+
| uID | plugin      | downloadDate |
+----+-----+-----+
| 4   | Bitcrush    | 2021-07-25 00:00:00 |
| 6   | Convolver    | 2021-10-09 00:00:00 |
| 7   | Distortion    | 2021-08-02 00:00:00 |
| 2   | Gate         | 2021-10-02 00:00:00 |
| 4   | Haas         | 2021-06-01 00:00:00 |
| 5   | Haas         | 2021-08-09 00:00:00 |
| 6   | Haas         | 2021-06-03 00:00:00 |
| 7   | Haas         | 2021-09-26 00:00:00 |
| 4   | Multipass    | 2021-09-23 00:00:00 |
| 6   | Multipass    | 2021-07-28 00:00:00 |
| 2   | PhasePlant   | 2021-09-03 00:00:00 |
| 1   | Pro-Q 3     | 2021-09-02 00:00:00 |
| 3   | Reverb        | 2021-08-27 00:00:00 |
+----+-----+-----+
13 rows in set (0.01 sec)
```

```
mysql> select * from User;
+----+-----+-----+-----+
| uID | name    | age   | downLoaded |
+----+-----+-----+-----+
| 1   | Kim     | 20    | 1          |
| 2   | John    | 24    | 2          |
| 3   | Amy     | 20    | 1          |
| 4   | Lucas   | 28    | 3          |
| 5   | Lio     | 19    | 1          |
| 6   | Mike    | 30    | 3          |
| 7   | Mia     | 21    | 2          |
+----+-----+-----+-----+
7 rows in set (0.01 sec)
```

plugin	developer	type
EQ	Kilohearts	FX
haper	XFER	FX
veling Tool	ADHD	FX
Ample Bass P Lite 2	Ample Sound	Generator
Ample Guitar M Lite	Ample Sound	Generator
h	Kilohearts	FX
Sweet V3	Flux	FX
nsweep V2	Plugin Alliance	FX
rack V3 Player	Plugin Alliance	FX
usher	Camel Audio	FX
	Kilohearts	FX
lter	Kilohearts	FX
sor	Kilohearts	FX
er	Kilohearts	FX
meter	Voxengo	FX
	Kilohearts	FX
Delay	Lama	Singing Monk
odulator	XFER	Generator
Daxed	Pascal Gauthier	Generator
er	Kilohearts	FX
ion	Kilohearts	FX
s	Kilohearts	FX
e	Kilohearts	FX
or	Kilohearts	FX
	Kilohearts	FX
Filter	Kilohearts	FX
ir	Slate Digital	FX
	Kilohearts	FX
	Kilohearts	FX
	GVST	FX
	dblue	FX
	GVST	FX
	Kilohearts	FX
Keyzone	Classic	Bitsonic Audio
	Tritik	Generator
LABS	Spitfire Audio	Generator
LABS Piano	Spitfire Audio	Generator

Filter	Kilohearts	FX
s Meter II	Kilohearts	FX
s Meter II Pro	Youlean	FX
n	Youlean	FX
tch	Melda Production	FX
ss	Melda Production	FX
	Melda Production	FX
ssor	Melda Production	FX
utionEZ	Melda Production	FX
e Essential	Celemony	FX
zer	Melda Production	FX
r	Melda Production	FX
ifter	Melda Production	FX
	Melda Production	FX
dulator	Melda Production	FX
tor	Melda Production	FX
o	Melda Production	FX
ss	Kilohearts	FX
o	Melda Production	FX
lder	Melda Production	FX
Filter	Plugin Alliance	FX
ar Filter	Kilohearts	FX
Verb	Voxengo	FX
	XFER	FX
Advanced	iZotope	FX
mager II	iZotope	FX
II	Cableguys	FX
Papu	Socalabs	Generator
istortion	Kilohearts	FX
PhasePlant	Kilohearts	Generator
	Kilohearts	FX
hifter	Kilohearts	FX
Podolski	u-he	Generator
	FabFilter	FX
or	Kilohearts	FX

		Kilohearts	FX
r		Kilohearts	FX
d		Kilohearts	FX
ider 3		Audio Damage	FX
ion Knob		Softube	FX
2		FabFilter	FX
Serum		XFER	Generator
Sforzando		Plogue	Generator
Q		Kilohearts	FX
p		Kilohearts	FX
		Voxengo	FX
		Kilohearts	FX
Synth1		Daichi Laboratory	Generator
erb 4		TAL	FX
TAL-BassLine		TAL	Generator
oder		TAL	FX
op		Kilohearts	FX
ay		ConcreteFX	FX
elnikov		Tokyo Dawn Records	FX
otok		Tokyo Dawn Records	FX
A		Tokyo Dawn Records	FX
SlickEQ		Tokyo Dawn Records	FX
		Creative Intent	FX
s 3		FabFilter	FX
Gate		Kilohearts	FX
nt Shaper		Kilohearts	FX
TyrellN6		u-he	Generator
a FreqEcho		Valhalla DSP	FX
a SpaceModulator		Valhalla DSP	FX
a SuperMassive		Valhalla DSP	FX
a UberMod		Valhalla DSP	FX
a VintageVerb		Valhalla DSP	FX
aDelay		Valhalla DSP	FX
aPlate		Valhalla DSP	FX
aRoom		Valhalla DSP	FX
aShimmer		Valhalla DSP	FX
VCV Rack		VCV Rack	Generator
		iZotope	FX
Vital		Matt Tytel	Generator
imiter no.6		Tokyo Dawn Records	FX
oublier		iZotope	FX
		VOLOCO	FX
Zebralette		u-he	Generator

```
+-----+-----+-----+
125 rows in set (0.0147 sec)
MySQL localhost:3306 ssl plugins SQL >
```

**7. List of at least 15 distinct functions excluding functions done by DBA. List each function in English and include all associated SQL statements, triggers, and stored procedures to support that particular function.**

- List all SQL select statements: 9 select queries

```
// select 1
// this select query will find developers who make both fx plugins and generator plugins and find
// total plugins each of these selected developers create.
    select developer, count(developer) as numOfPlugins from plugins old
    where type !=any (select type from plugins where developer = old.developer)
group by (developer);

// select 2
// this select query will find the type of plugin that is downloaded the most by users
    select type from plugins join (
        select plugin, count from (
            select plugin, count(plugin) as count from downloaded group by plugin) as d1
        where d1.count >= all(
            select count(plugin) from downloaded group by plugin))d2
        where d2.plugin = plugins.plugin;

// select 3
// this query will find users who downloaded both fx and generator plugins
    select name from user where uid = (select uID from
        (select uID, plugin from downloaded natural join plugins
        where plugin != all (select name from fx))t1 join
        (select uID, plugin from downloaded natural join plugins
        where plugin = any (select name from fx))t2 using(uID));

//select 4
//select the names of the plugins that have been downloaded from Fx and Generator after the
//year 2020
select distinct generator.name from generator
left outer join downloaded on generator.name = downloaded.plugin
where generator.lastUpdate > '2019-12-31' and downloaded.downloadDate > '2019-12-31'
union all
select distinct fx.name from fx
right outer join downloaded on fx.name = downloaded.plugin
where fx.lastUpdate > '2019-12-31' and downloaded.downloadDate > '2019-12-31';
```

```

//select 5
//select number of generator plugins and fx plugins in VST3 format
select count(distinct generator.name) as genCount, count(distinct fx.name) as fxCount
from generator, FX
where generator.format = 'VST3' and fx.format = 'VST3';

//select 6
//select average number of plugins in fx table that are free
select avg(stcounts.ct)
from(
select count(subtype) as ct
from fx
where fx.isFree = 1
group by subtype
having count(subtype) > 1
) as stcounts;

//select 7
//select all users that have the same number of downloads
select distinct u1.name as User1, u2.name as User2
from User u1, User u2
where u1.name < u2.name and u1.downLoaded = u2.downLoaded
group by u1.uID;

//select 8 find developers made only one type of plugins and not the other and the total the
//plugins that developer made
select developer, type, count(developer) as numOfPlugins from plugins where developer != any
(select developer from plugins old where type != any (
select type from plugins where old.developer = developer) group by developer) group by
developer;

// select 9 : select user whose download is not free but has license
select user.name from (select uid from (select * from downloaded left outer join generator on
plugin = name union select * from downloaded left outer join fx on plugin = name) t where
t.name is not null and isFree = 0 and hasLicense = 1 group by(uid)) s natural join user;

```

- List all SQL update statements: 2 update queries

```

// update 1: update the plugin 'Delay Lama' with the current time.
update generator set lastUpdate = now() where name = 'Delay Lama';

```

```
// update 2 : update generator by inverting the isFree variable for plugins with the subtype  
'ROMpler'
```

```
update generator set isFree = not isFree where subtype = 'ROMpler';
```

- List all SQL delete statements: 2 delete queries

```
// delete 1 : work with trigger 2
```

```
/// delete the uid =7, and plugin = 'Haas' and downloadDate='2021-09-26' will delete the record  
// from the downloaded table as well as update the user table for the downloaded times for uid 7  
// in this case if uid has downloaded 2 times, then after deletion from downloaded table, his/her  
// user table will be updated to 1 times (work with trigger 2)
```

```
delete from downloaded where uid = 7 and plugin ='Haas' and downloadDate='2021-09-26';
```

```
// delete 2 // expiration after 4 months
```

```
//delete plugins that have not been updated in 4 months
```

```
delete from fx
```

```
where NOW() > DATE_ADD(fx.lastUpdate, INTERVAL 4 MONTH);
```

- List all SQL insert statements: 2 insert queries

```
// insert 1: work with trigger 1
```

```
// for insert into downloaded table where user download new plugins at current time  
insert into downloaded values(1, 'Haas', now());
```

```
// insert 2: insert a new user
```

```
insert into user(name, age, downloaded) values('Anna', 20, 0);
```

- List all SQL triggers: 2 triggers

```
// trigger 1:
```

```
// create trigger validateUserForDownload before insert into downloaded table, need to check  
//if user is an existing user, and will add 1 to total download times for that user in the user table.  
delimiter //
```

```
drop trigger if exists validateUserForDownload;
```

```
create trigger validateUserForDownload
```

```
before insert on downloaded
```

```
for each row
```

```
if new.uID in (select uID from user)
```

```
then update User set downLoaded = downLoaded +1 where uID = new.uID;
```

```
end if; //
```

```
delimiter ;
```

```

// trigger 2:
// delete record from downloaded table, the user table downloaded plugin's time will be reduced
// as well, this trigger will not check delete record input for validation. So it is allow delete
//multiple record, as long as the value in the where clause exist in the downloaded table
delimiter //
drop trigger if exists deleteDownloadRecord;
create trigger deleteDownloadRecord
before delete on downloaded
for each row
begin
update User set downloaded = downloaded -1 where uid = old.uid;
end;//
delimiter ;

```

- List all SQL stored procedures: 1 stored procedures

```

// stored procedure table archive will store old plugins that its lastUpdate was 2020 and earlier.
//this procedure will first insert into archive using data from plugins join fx and plugins join
//generator using common attribute plugin name and the given parameter 'last_update' . Then
//delete old plugs info from plugins, fx and generator table using common attributes and given
//parameters. Archive table also include user who has downloaded that plugins

```

```

drop procedure if exists archive;
delimiter //
create procedure archive(IN last_update timestamp)
begin
insert into archive (name, format, developer, type, subtype, isFree, hasLicense, lastUpdate, uID,
downloadDate) select name, format, developer, type, subtype, isFree, hasLicense, lastUpdate,
uID, downloadDate from plugins join fx on fx.name = plugins.plugin and fx.lastUpdate <
last_update left outer join downloaded on downloaded.plugin = plugins.plugin;
insert into archive (name, format, developer, type, subtype, isFree, hasLicense, lastUpdate, uID,
downloadDate) select name, format, developer, type, subtype, isFree, hasLicense, lastUpdate,
uID, downloadDate from plugins join generator on generator.name = plugins.plugin and
generator.lastUpdate < last_update left outer join downloaded on downloaded.plugin =
plugins.plugin;
delete from plugins where plugins.plugin in (select name from fx where lastUpdate <
last_update);
delete from plugins where plugins.plugin in (select name from generator where lastUpdate <
last_update);
delete from fx where lastUpdate < last_update;
delete from generator where lastUpdate < last_update;

```

```
end //  
delimiter ;  
  
call archive('2020-01-01');
```

8. The following requirements are for me to check there are at least 5 significantly different queries involving different relations and attributes.

- Please highlight in bold the five significantly different types of SQL (one co-related subquery, group by and having, aggregation, outer join, and mathematical set operation)
- List all SQL select statements: 9 select queries

```
// select 1
// this select query will find developers who make both fx plugins and generator plugins and find
// total plugins each of these selected developers create.
    select developer, count(developer) as numOfPlugins from plugins old
    where type !=any (select type from plugins where developer = old.developer)
group by (developer);

// select 2
// this select query will find the type of plugin that is downloaded the most by users
    select type from plugins join (
        select plugin, count from (
            select plugin, count(plugin) as count from downloaded group by plugin) as d1
        where d1.count >= all (
            select count(plugin) from downloaded group by plugin))d2
        where d2.plugin = plugins.plugin;

// select 3
// this query will find users who downloaded both fx and generator plugins
    select name from user where uid = (select uID from
        (select uID, plugin from downloaded natural join plugins
        where plugin != all (select name from fx)) t1 join
        (select uID, plugin from downloaded natural join plugins
        where plugin = any (select name from fx)) t2 using(uID));

//select 4
//select the names of the plugins that have been downloaded from Fx and Generator after the
//year 2020

select distinct generator.name from generator
left outer join downloaded on generator.name = downloaded.plugin
where generator.lastUpdate > '2019-12-31' and downloaded.downloadDate > '2019-12-31'
union all
select distinct fx.name from fx
right outer join downloaded on fx.name = downloaded.plugin
```

```

where fx.lastUpdate > '2019-12-31' and downloaded.downloadDate > '2019-12-31';

//select 5
//select number of generator plugins and fx plugins in VST3 format
select count(distinct generator.name) as genCount, count(distinct fx.name) as fxCount
from generator, FX
where generator.format = 'VST3' and fx.format = 'VST3';

//select 6
//select average number of plugins in fx table that are free
select avg(stcounts.ct)
from(
select count(subtype) as ct
from fx
where fx.isFree = 1
group by subtype
having count(subtype) > 1
) as stcounts;

//select 7
//select all users that have the same number of downloads
select distinct u1.name as User1, u2.name as User2
from User u1, User u2
where u1.name < u2.name and u1.downLoaded = u2.downLoaded
group by u1.uID;

//select 8 find developers made only one type of plugins and not the other and the total the
//plugins that developer made
select developer, type, count(developer) as numOfPlugins from plugins where developer != any
(select developer from plugins old where type != any (
select type from plugins where old.developer = developer) group by developer) group by
developer;

// select 9 : select user whose download is not free but has license
select user.name from (select uid from (select * from downloaded left outer join generator on
plugin = name union select * from downloaded left outer join fx on plugin = name) t where
t.name is not null and isFree = 0 and hasLicense = 1 group by(uid)) s natural join user;



- List all SQL update statements: 2 update queries


// update 1: update the plugin 'Delay Lama' with the current time.

```

```

update generator set lastUpdate = now() where name = 'Delay Lama';

// update 2 : update generator by inverting the isFree variable for plugins with the subtype
'ROMpler'

update generator

set isFree = not isFree

where subtype = 'ROMpler';



- List all SQL delete statements: 2 delete queries



// delete 1 : work with trigger 2
/// delete the uid =7, and plugin = 'Haas' and downloadDate='2021-09-26' will delete the record
// from the downloaded table as well as update the user table for the downloaded times for uid 7
delete from downloaded where uid = 7 and plugin ='Haas' and downloadDate='2021-09-26';

// delete 2 // expiration after 4 months
//delete plugins that have not been updated in 4 months
delete from fx
where NOW() > DATE_ADD(fx.lastUpdate, INTERVAL 4 MONTH);



- List all SQL insert statements: 2 insert queries



// insert 1: work with trigger 1
// for insert into downloaded table where user download new plugins at current time
insert into downloaded values(1, 'Haas', now());

// insert 2: insert a new user
insert into user(name, age, downloaded) values('Anna', 20, 0);



- List all SQL triggers: 2 triggers



// trigger 1:
// create trigger validateUserForDownload before insert into downloaded table, need to check
// if user is an existing user, and will add 1 to total download times for that user in the user table.
delimiter //
drop trigger if exists validateUserForDownload;
create trigger validateUserForDownload
before insert on downloaded
for each row

```

```

if new.uID in (select uID from user)
then update User set downLoaded = downLoaded +1 where uID = new.uID;
end if;//
delimiter ;

// trigger 2:
// delete record from downloaded table, the user table downloaded plugin's time will be reduced
// as well, this trigger will not check delete record input for validation. So it is allow delete
//multiple record, as long as the value in the where clause exist in the downloaded table
delimiter //
drop trigger if exists deleteDownloadRecord;
create trigger deleteDownloadRecord
before delete on downloaded
for each row
begin
update User set downloaded = downloaded -1 where uid = old.uid;
end;//
delimiter ;

```

- List all SQL stored procedures: 1 stored procedures

```

// stored procedure table archive will store old plugins that its lastUpdate was 2020 and earlier.
//this procedure will first insert into archive using data from plugins join fx and plugins join
//generator using common attribute plugin name and the given parameter ‘last_update’ . Then
//delete old plugs info from plugins, fx and generator table using common attributes and given
//parameters. Archive table also include user who has downloaded that plugins

```

```

drop procedure if exists archive;
delimiter //
create procedure archive(IN last_update timestamp)
begin
insert into archive (name, format, developer, type, subtype, isFree, hasLicense, lastUpdate, uID,
downloadDate) select name, format, developer, type, subtype, isFree, hasLicense, lastUpdate,
uID, downloadDate from plugins join fx on fx.name = plugins.plugin and fx.lastUpdate <
last_update left outer join downloaded on downloaded.plugin = plugins.plugin;
insert into archive (name, format, developer, type, subtype, isFree, hasLicense, lastUpdate, uID,
downloadDate) select name, format, developer, type, subtype, isFree, hasLicense, lastUpdate,
uID, downloadDate from plugins join generator on generator.name = plugins.plugin and
generator.lastUpdate < last_update left outer join downloaded on downloaded.plugin =
plugins.plugin;
delete from plugins where plugins.plugin in (select name from fx where lastUpdate <
last_update);

```

```
delete from plugins where plugins.plugin in (select name from generator where lastUpdate < last_update);
delete from fx where lastUpdate < last_update;
delete from generator where lastUpdate < last_update;
end //
delimiter ;
```

  

```
call archive('2020-01-01');
```

**9. Screenshots to demonstrate the following functionality. Label each screenshot clearly.**

- o each 15 functions
- o archiving
- o key constraint and foreign key constraint violations

```
// select 1
// this select query will find developers who make both fx plugins and generator plugins and find
// total plugins each of these selected developers create.
```

Tables loaded!

developer	numOfPlugins	
Kilohearts	37	
XFER	4	
TAL	3	

```
MySQL [localhost:3306 ssl] plugins SQL > select developer, count(developer) as numOfPlugins from plugins old
                                         ->      where type != any (select type from plugins where developer = old.devel
oper)
                                         ->      group by (developer);
+-----+-----+
| developer | numOfPlugins |
+-----+-----+
| Kilohearts |      37 |
| XFER       |       4 |
| TAL        |       3 |
+-----+-----+
3 rows in set (0.0084 sec)
```

```
// select 2
// this select query will find the type of plugin that is downloaded the most by users
```

Tables loaded!	
	type
	FX

```

mysql> select type from plugins join (
->     select plugin, count from (
->         select plugin, count(plugin) as count from downloaded group by plugin) as d1
->         where d1.count >= all(
->             select count(plugin) from downloaded group by plugin))d2
->             where d2.plugin = plugins.plugin;
+-----+
| type |
+-----+
| X
+-----+
1 row in set (0.01 sec)

```

// select 3

// this query will find users who downloaded both fx and generator plugins

Tables loaded!	
	name
	John

```

mysql> select name from user where uid = (select uID from
->     (select uID, plugin from downloaded natural join plugins
->     where plugin != all (select name from fx))t1 join
->     (select uID, plugin from downloaded natural join plugins
->     where plugin = any (select name from fx))t2 using(uID));
+-----+
| name |
+-----+
| John |
+-----+
1 row in set (0.00 sec)

```

//select 4

//select the names of the plugins that have been downloaded from Fx and Generator after the  
//year 2020

Tables loaded!	
	name
	PhasePlant
	Bitcrush
	Convolver
	Distortion
	Gate
	Haas
	Multipass
	Reverb

```
//select 5  
//select number of generator plugins and fx plugins in VST3 format
```

Tables loaded!

genCount	fxCount	
5	90	

```
MySQL [localhost:3306 ssl] [plugins] [SQL] > select count(distinct generator.name) as genCount, count(distinct fx.name) as fxCount  
                                         -> from generator, FX  
                                         -> where generator.format = 'VST3' and fx.format = 'VST3';  
+-----+-----+  
| genCount | fxCount |  
+-----+-----+  
|      5   |     90  |  
+-----+-----+  
1 row in set (0.0032 sec)
```

```
//select 6  
//select average number of plugins in fx table that are free
```

Tables loaded!

avg(stcounts.ct)	
4.9091	

```
MySQL [localhost:3306 ssl] [plugins] [SQL] > select avg(stcounts.ct)  
                                         -> from(  
                                         -> select count(subtype) as ct  
                                         -> from fx  
                                         -> where fx.isFree = 1  
                                         -> group by subtype  
                                         -> having count(subtype) > 1  
                                         -> ) as stcounts;  
+-----+  
| avg(stcounts.ct) |  
+-----+  
|      4.9091    |  
+-----+  
1 row in set (0.0011 sec)
```

```
//select 7  
//select all users that have the same number of downloads
```

Tables loaded!

name	name
Amy	Kim
John	Mia
Kim	Lio
Lucas	Mike

```
MySQL localhost:3306 ssl plugins SQL > select distinct u1.name as User1, u2.name as User2
-> from User u1, User u2
-> where u1.name < u2.name and u1.downLoaded = u2.downLoaded
-> group by u1.uID;

+-----+-----+
| User1 | User2 |
+-----+-----+
| Amy   | Kim   |
| John  | Mia   |
| Kim   | Lio   |
| Lucas | Mike  |
+-----+-----+
4 rows in set (0.0025 sec)
```

//select 8 find developers made only one type of plugins and not the other and the total the  
//plugins that developer made

Tables loaded!		
developer	type	numOfPlugins
Kiloharts	FX	37
XFER	FX	4
ADHD	FX	1
Ample Sound	Generator	2
Flux	FX	1
Plugin Alliance	FX	3
Camel Audio	FX	1
Voxengo	FX	3
Singing Monk	Generator	1
Pascal Gauthier	Generator	1
Slate Digital	FX	1
GVST	FX	2
dblue	FX	1
Bitsonic Audio	Generator	1

Tables loaded!		
developer	type	numOfPlugins
Tritik	FX	1
Spitfire Audio	Generator	2
Youlean	FX	2
Melda Production	FX	16
Celemony	FX	1
iZotope	FX	4
CableGuys	FX	1
SocaLabs	Generator	1
u-he	Generator	3
FabFilter	FX	9
Audio Damage	FX	1
Softube	FX	1
Plogue	Generator	1
Daichi Laboratory	Generator	1

TAL	FX	3
ConcreteFX	FX	1
Tokyo Dawn Reco...	FX	5
Creative Intent	FX	1
Valhalla DSP	FX	9
VCV Rack	Generator	1
Matt Tytel	Generator	1
VOLOCO	FX	1

```
MySQL [localhost:3306 ssl plugins SQL] > select developer, type, count(developer) as numOfPlugins from plugins where developer != any (select developer from plugin old where type != any (
    select type from plugins where old.developer = developer) group by developer) group by developer;
+-----+-----+-----+
| developer | type | numOfPlugins |
+-----+-----+-----+
|          | X    | 37           |
|          | X    | 4             |
|          | X    | 1             |
|          | Generator | 2             |
|          | X    | 1             |
|          | X    | 3             |
|          | X    | 1             |
|          | X    | 3             |
|          | Generator | 1             |
|          | Generator | 1             |
|          | X    | 1             |
|          | X    | 2             |
|          | X    | 1             |
|          | Generator | 1             |
|          | X    | 1             |
|          | Generator | 2             |
|          | X    | 16            |
|          | X    | 1             |
|          | X    | 4             |
|          | X    | 1             |
|          | Generator | 1             |
|          | Generator | 3             |
|          | X    | 9             |
|          | X    | 1             |
|          | X    | 1             |
|          | X    | 1             |
|          | Generator | 1             |
|          | Generator | 1             |
|          | X    | 3             |
|          | X    | 1             |
|          | X    | 5             |
|          | X    | 1             |
|          | X    | 9             |
|          | Generator | 1             |
|          | Generator | 1             |
|          | X    | 1             |
+-----+-----+-----+
36 rows in set (0.0106 sec)
MySQL [localhost:3306 ssl plugins SQL] >
```

// select 9 : select user whose download is not free but has license

Tables loaded!

developer	type	numOfPlugins
Kilohearts	FX	37
XFER	FX	4
ADiHD	FX	1
Ample Sound	Generator	2
Flux	FX	1
Plugin Alliance	FX	3
Camel Audio	FX	1
Voxengo	FX	3
Singing Monk	Generator	1
Pascal Gauthier	Generator	1
Slate Digital	FX	1
GVST	FX	2
dblue	FX	1
Bitsonic Audio	Generator	1
Trinik	FX	1
Spitfire Audio	Generator	2
Youlean	FX	2
Melda Production	FX	16
Celemony	FX	1
iZotope	FX	4
CableGuys	FX	1
SocaLabs	Generator	1
u-he	Generator	3
FabFilter	FX	9
Audio Damage	FX	1
Softube	FX	1
Plogue	Generator	1
Dalchi Laboratory	Generator	1
TAL	FX	3
ConcreteFX	FX	1
Tokyo Dawn Reco...	FX	5
Creative Intent	FX	1
Valhalla DSP	FX	9
VCV Rack	Generator	1
Matt Tytel	Generator	1
VOLOCO	FX	1

```

MySQL [localhost:3306 ssl] plugins SQL> select developer, type, count(developer) as numOfPlugins
      from plugins
      where developer != any (select developer from plugins old where type != any
                                (select type from plugins where old.developer = developer) group by developer)
      group by developer, type;
+-----+-----+-----+
| developer | type | numOfPlugins |
+-----+-----+-----+
| 37 | X | 1 |
| 4 | X | 1 |
| 1 | X | 1 |
| 2 | Generator | 1 |
| 1 | X | 1 |
| 3 | X | 1 |
| 1 | X | 1 |
| 3 | X | 1 |
| 1 | Generator | 1 |
| 1 | Generator | 1 |
| 1 | X | 1 |
| 2 | X | 1 |
| 1 | X | 1 |
| 1 | Generator | 1 |
| 1 | X | 1 |
| 2 | Generator | 2 |
| 2 | X | 1 |
| 16 | X | 1 |
| 1 | X | 1 |
| 4 | X | 1 |
| 1 | X | 1 |
| 1 | Generator | 1 |
| 1 | Generator | 1 |
| 3 | X | 1 |
| 9 | X | 1 |
| 1 | X | 1 |
| 1 | X | 1 |
| 1 | Generator | 1 |
| 1 | Generator | 1 |
| 1 | X | 1 |
| 3 | X | 1 |
| 1 | X | 1 |
| 5 | X | 1 |
| 1 | X | 1 |
| 9 | X | 1 |
| 1 | Generator | 1 |
| 1 | Generator | 1 |
| 1 | X | 1 |
+-----+-----+-----+
36 rows in set (0.0109 sec)
MySQL [localhost:3306 ssl] plugins SQL> 
```

```
// update 1: update the plugin 'Delay Lama' with the current time.
```

## Before update:

Tables loaded!						
<b>name</b>	<b>format</b>	<b>subtype</b>	<b>isFree</b>	<b>hasLicense</b>	<b>lastUpdate</b>	
Delay Lama	VST2	Softsynth	1	0	2021-12-04 20:06:15	

```
MySQL localhost:3306 ssl plugins SQL > select * from generator where name = 'Delay Lama';

+-----+-----+-----+-----+-----+
| name | format | subtype | isFree | hasLicense | lastUpdate |
+-----+-----+-----+-----+-----+
| Delay Lama | VST2 | Softsynth | 1 | 0 | 2021-12-04 20:06:15 |
+-----+-----+-----+-----+-----+
1 row in set (0.0009 sec)
```

After update:

Tables loaded!						
<b>name</b>	<b>format</b>	<b>subtype</b>	<b>isFree</b>	<b>hasLicense</b>	<b>lastUpdate</b>	
Delay Lama	VST2	Softsynth	1	0	2021-12-04 21:02:13	

```
MySQL localhost:3306 ssl plugins SQL > select * from generator where name = 'Delay Lama';

+-----+-----+-----+-----+-----+
| name | format | subtype | isFree | hasLicense | lastUpdate |
+-----+-----+-----+-----+-----+
| Delay Lama | VST2 | Softsynth | 1 | 0 | 2021-12-04 21:02:13 |
+-----+-----+-----+-----+-----+
1 row in set (0.0026 sec)
```

// update 2 : update generator by inverting the isFree variable for plugins with the subtype 'ROMpler'

Before update:

Tables loaded!

name	format	subtype	isFree	hasLicense	lastUpdate
Ample Bass P Lite 2	VST2	ROMpler	0	0	2021-12-04 20:02:26
Ample Guitar M Lite	VST2	ROMpler	0	0	2021-12-04 20:02:26
Keyzone Classic	VST2	ROMpler	0	0	2021-12-04 20:02:26
LABS	VST3	ROMpler	0	0	2021-12-04 20:02:26
LABS Piano	VST3	ROMpler	0	0	2021-12-04 20:02:26
Sforzando	VST2	ROMpler	0	0	2021-12-04 20:02:26

```
MySQL localhost:3306 ssl plugins SQL > select * from generator where subtype = 'ROMpler';

+-----+-----+-----+-----+-----+
| name | format | subtype | isFree | hasLicense | lastUpdate |
+-----+-----+-----+-----+-----+
| Ample Bass P Lite 2 | VST2 | ROMpler | 0 | 0 | 2021-12-04 21:04:41 |
| Ample Guitar M Lite | VST2 | ROMpler | 0 | 0 | 2021-12-04 21:04:41 |
| Keyzone Classic | VST2 | ROMpler | 0 | 0 | 2021-12-04 21:04:41 |
| LABS | VST3 | ROMpler | 0 | 0 | 2021-12-04 21:04:41 |
| LABS Piano | VST3 | ROMpler | 0 | 0 | 2021-12-04 21:04:41 |
| Sforzando | VST2 | ROMpler | 0 | 0 | 2021-12-04 21:04:41 |
+-----+-----+-----+-----+-----+
6 rows in set (0.0026 sec)
```

After update:

Tables loaded!

name	format	subtype	isFree	hasLicense	lastUpdate
Ample Bass P Lite 2	VST2	ROMpler	1	0	2021-12-04 20:07:15
Ample Guitar M Lite	VST2	ROMpler	1	0	2021-12-04 20:07:15
Keyzone Classic	VST2	ROMpler	1	0	2021-12-04 20:07:15
LABS	VST3	ROMpler	1	0	2021-12-04 20:07:15
LABS Piano	VST3	ROMpler	1	0	2021-12-04 20:07:15
Sforzando	VST2	ROMpler	1	0	2021-12-04 20:07:15

```

MySQL [localhost:3306 ssl] plugins SQL> select * from generator where subtype = 'ROMpler';

+-----+-----+-----+-----+-----+-----+
| name          | format | subtype | isFree | hasLicense | lastUpdate |
+-----+-----+-----+-----+-----+-----+
| Ample Bass P Lite 2 | VST2   | ROMpler | 1      | 0           | 2021-12-04 20:07:15 |
| Ample Guitar M Lite | VST2   | ROMpler | 1      | 0           | 2021-12-04 20:07:15 |
| Keyzone Classic    | VST2   | ROMpler | 1      | 0           | 2021-12-04 20:07:15 |
| LABS              | VST3   | ROMpler | 1      | 0           | 2021-12-04 20:07:15 |
| LABS Piano         | VST3   | ROMpler | 1      | 0           | 2021-12-04 20:07:15 |
| Sforzando          | VST2   | ROMpler | 1      | 0           | 2021-12-04 20:07:15 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.0009 sec)

```

```

// delete 1 : work with trigger 2
/// delete the uid =7, and plugin = 'Haas' and downloadDate='2021-09-26' will delete the record
// from the downloaded table as well as update the user table for the downloaded times for uid 7

```

Before delete:

Tables loaded!			
uID	plugin	downloadDate	
4	Bitcrush	2021-07-25 00:00:00	
6	Convolver	2021-10-09 00:00:00	
7	Distortion	2021-08-02 00:00:00	
2	Gate	2021-10-02 00:00:00	
4	Haas	2021-06-01 00:00:00	
5	Haas	2021-08-09 00:00:00	
6	Haas	2021-06-03 00:00:00	
7	Haas	2021-09-26 00:00:00	
4	Multipass	2021-09-23 00:00:00	
6	Multipass	2021-07-28 00:00:00	
2	PhasePlant	2021-09-03 00:00:00	
1	Pro-Q 3	2021-09-02 00:00:00	
3	Reverb	2021-08-27 00:00:00	

Tables loaded!

uID	name	age	downLoaded
1	Kim	20	2
2	John	24	4
3	Amy	20	2
4	Lucas	28	6
5	Lio	19	2
6	Mike	30	6
7	Mia	21	4

```
mysql> select * from downloaded;
+----+-----+-----+
| uID | plugin      | downloadDate          |
+----+-----+-----+
| 4  | Bitcrush    | 2021-07-25 00:00:00 |
| 6  | Convolver   | 2021-10-09 00:00:00 |
| 7  | Distortion   | 2021-08-02 00:00:00 |
| 2  | Gate         | 2021-10-02 00:00:00 |
| 4  | Haas         | 2021-06-01 00:00:00 |
| 5  | Haas         | 2021-08-09 00:00:00 |
| 6  | Haas         | 2021-06-03 00:00:00 |
| 7  | Haas         | 2021-09-26 00:00:00 |
| 4  | Multipass    | 2021-09-23 00:00:00 |
| 6  | Multipass    | 2021-07-28 00:00:00 |
| 2  | PhasePlant   | 2021-09-03 00:00:00 |
| 1  | Pro-Q 3      | 2021-09-02 00:00:00 |
| 3  | Reverb        | 2021-08-27 00:00:00 |
+----+-----+-----+
13 rows in set (0.00 sec)

mysql> select * from user;
+----+-----+-----+
| uID | name      | age | downloaded |
+----+-----+-----+
| 1  | Kim       | 20  | 2           |
| 2  | John      | 24  | 4           |
| 3  | Amy       | 20  | 2           |
| 4  | Lucas     | 28  | 6           |
| 5  | Lio       | 19  | 2           |
| 6  | Mike      | 30  | 6           |
| 7  | Mia       | 21  | 4           |
+----+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

After delete:

Tables loaded!

uID	plugin	downloadDate
4	Bitcrush	2021-07-25 00:00:00
6	Convolver	2021-10-09 00:00:00
7	Distortion	2021-08-02 00:00:00
2	Gate	2021-10-02 00:00:00
4	Haas	2021-06-01 00:00:00
5	Haas	2021-08-09 00:00:00
6	Haas	2021-06-03 00:00:00
4	Multipass	2021-09-23 00:00:00
6	Multipass	2021-07-28 00:00:00
2	PhasePlant	2021-09-03 00:00:00
1	Pro-Q 3	2021-09-02 00:00:00
3	Reverb	2021-08-27 00:00:00

Tables loaded!

uID	name	age	downLoaded
1	Kim	20	2
2	John	24	4
3	Amy	20	2
4	Lucas	28	6
5	Lio	19	2
6	Mike	30	6
7	Mia	21	3

```

mysql> select * from downloaded;
+-----+-----+-----+
| uID | plugin      | downloadDate          |
+-----+-----+-----+
| 4   | Bitcrush    | 2021-07-25 00:00:00 |
| 6   | Convolver    | 2021-10-09 00:00:00 |
| 7   | Distortion   | 2021-08-02 00:00:00 |
| 2   | Gate         | 2021-10-02 00:00:00 |
| 4   | Haas         | 2021-06-01 00:00:00 |
| 5   | Haas         | 2021-08-09 00:00:00 |
| 6   | Haas         | 2021-06-03 00:00:00 |
| 4   | Multipass    | 2021-09-23 00:00:00 |
| 6   | Multipass    | 2021-07-28 00:00:00 |
| 2   | PhasePlant   | 2021-09-03 00:00:00 |
| 1   | Pro-Q 3     | 2021-09-02 00:00:00 |
| 3   | Reverb        | 2021-08-27 00:00:00 |
+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> select * from user;
+-----+-----+-----+-----+
| uID | name       | age    | downLoaded |
+-----+-----+-----+-----+
| 1   | Kim         | 20    | 2           |
| 2   | John        | 24    | 4           |
| 3   | Amy         | 20    | 2           |
| 4   | Lucas       | 28    | 6           |
| 5   | Lio          | 19    | 2           |
| 6   | Mike         | 30    | 6           |
| 7   | Mia          | 21    | 3           |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

```

Note: deleting from downloaded affects user through trigger

```

// delete 2 // expiration after 4 months
//delete plugins that have not been updated in 4 months

```

Before delete:

Tables loaded!						
name	format	subtype	isFree	hasLicense	lastUpdate	
3-Band EQ	VST3	EQ	1	0	2020-11-28 00:00:00	
8 Bit Shaper	VST2	Distortion	1	0	2017-11-29 00:00:00	
ADHD Leveling Tool	VST2	Dynamics	1	0	2018-12-01 00:00:00	
Bitcrush	VST3	Distortion	0	1	2020-11-28 00:00:00	
Bitter Sweet V3	VST3	Dynamics	1	0	2019-04-11 00:00:00	
BX_Cleansweep V2	VST3	Filter	1	1	2017-05-24 00:00:00	
BX_Rockrack V3 Player	VST3	Distortion	1	1	2017-05-24 00:00:00	
CamerCrusher	VST2	Distortion	1	0	2018-12-01 00:00:00	
CarveEQ	VST3	EQ	0	1	2020-11-28 00:00:00	
Chorus	VST3	Chorus	1	0	2020-11-28 00:00:00	
Comb Filter	VST3	Filter	0	1	2020-11-28 00:00:00	
Compressor	VST3	Dynamics	0	1	2020-11-28 00:00:00	
Convolver	VST3	Reverb	0	1	2020-11-28 00:00:00	
Correlometer	VST2	Monitoring	1	0	2017-09-22 00:00:00	
Delay	VST3	Delay	1	0	2020-11-28 00:00:00	
Delta Modulator	VST2	Distortion	1	0	2017-11-29 00:00:00	
Disperser	VST3	Filter	0	1	2020-11-28 00:00:00	
Distortion	VST3	Distortion	0	1	2020-11-28 00:00:00	
Dynamics	VST3	Dynamics	0	1	2020-11-28 00:00:00	
Ensemble	VST3	Chorus	0	1	2020-11-28 00:00:00	
Faturator	VST3	Distortion	0	1	2020-11-28 00:00:00	
Filter	VST3	Filter	0	1	2020-11-28 00:00:00	
Flanger	VST3	Filter	0	1	2020-11-28 00:00:00	
Formant Filter	VST3	Filter	0	1	2020-11-28 00:00:00	

Fresh Air	VST3	Misc	1	0	2020-11-27 00:00:00
Gain	VST3	Utility	1	0	2020-11-28 00:00:00
Gate	VST3	Gate	0	1	2020-11-28 00:00:00
GClip	VST2	Distortion	1	0	2020-12-27 00:00:00
Glitch	VST2	Host	1	0	2019-04-11 00:00:00
GSnap	VST2	Pitch Correction	1	0	2018-01-17 00:00:00
Haas	VST3	Imaging	0	1	2020-11-28 00:00:00
Krush	VST3	Distortion	1	0	2017-06-19 00:00:00
Ladder Filter	VST3	Filter	0	1	2020-11-28 00:00:00
Limiter	VST3	Dynamics	1	0	2020-11-28 00:00:00
Loudness Meter II	VST3	Monitoring	1	0	2017-09-22 00:00:00
Loudness Meter II Pro	VST3	Monitoring	0	1	2018-09-20 00:00:00
MAutoPan	VST3	Modulation	1	0	2017-04-05 00:00:00
MAutoPitch	VST3	Pitch Correction	1	0	2017-04-05 00:00:00
MBandPass	VST3	Filter	1	0	2017-04-05 00:00:00
MbitFun	VST3	Distortion	1	0	2017-04-05 00:00:00
MComb	VST3	Filter	1	0	2017-04-05 00:00:00
MCompressor	VST3	Dynamics	1	0	2017-04-05 00:00:00
MConvolutionEZ	VST3	Reverb	1	0	2017-04-05 00:00:00
Melodyne Essential	VST3	Pitch Correction	0	1	2020-08-21 00:00:00
MEqualizer	VST3	EQ	1	0	2017-04-05 00:00:00
MFlanger	VST3	Filter	1	0	2017-04-05 00:00:00
MFreqShifter	VST3	Misc	1	0	2017-04-05 00:00:00
MPhaser	VST3	Filter	1	0	2017-04-05 00:00:00

MRingModulator	VST3	Modulation	1	0	2017-04-05 00:00:00
MSaturator	VST3	Distortion	1	0	2017-04-05 00:00:00
MTremolo	VST3	Modulation	1	0	2017-04-05 00:00:00
Multipass	VST3	Host	0	1	2020-11-28 00:00:00
MVibrato	VST3	Pitch Shift	1	0	2017-04-05 00:00:00
MWaveFolder	VST3	Distortion	1	0	2017-04-05 00:00:00
Niveau Filter	VST3	Filter	1	1	2017-05-24 00:00:00
Nonlinear Filter	VST3	Filter	0	1	2020-11-28 00:00:00
OldSkooolVerb	VST2	Reverb	0	0	2017-09-22 00:00:00
OTT	VST2	Dynamics	1	0	2017-11-29 00:00:00
Ozone 9 Advanced	VST3	Mastering	0	1	2020-08-21 00:00:00
Ozone Imager II	VST3	Imaging	1	1	2017-06-19 00:00:00
PanCake II	VST2	Modulation	1	0	2020-12-27 00:00:00
Phase Distortion	VST3	Distortion	0	1	2020-11-28 00:00:00
Phaser	VST3	Filter	0	1	2020-11-28 00:00:00
Pitch Shifter	VST3	Pitch Shift	0	1	2020-11-21 00:00:00
Pro-C 2	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-DS	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-G	VST3	Gate	0	1	2021-08-08 00:00:00
Pro-L 2	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-MB	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-Q 3	VST3	EQ	0	1	2019-04-30 00:00:00
Pro-R	VST3	Reverb	0	1	2020-11-25 00:00:00
Resonator	VST3	Misc	0	1	2020-11-28 00:00:00

Reverb	VST3	Reverb	0	1	2020-11-28 00:00:00
Reverser	VST3	Misc	0	1	2020-11-28 00:00:00
Ring Mod	VST3	Modulation	0	1	2020-11-28 00:00:00
Rough Rider 3	VST2	Dynamics	1	0	2018-10-08 00:00:00
Saturation Knob	VST3	Distortion	1	1	2017-06-19 00:00:00
Saturn 2	VST3	Distortion	0	1	2021-08-08 00:00:00
Slice EQ	VST3	EQ	0	1	2020-11-28 00:00:00
SnapHeap	VST3	Host	0	0	2020-11-28 00:00:00
SPAN	VST3	Monitoring	1	0	2017-09-22 00:00:00
Stereo	VST3	Imaging	1	0	2020-11-28 00:00:00
TAL Reverb 4	VST2	Reverb	1	0	2020-03-19 00:00:00
TAL-Vocoder	VST2	Misc	1	0	2017-04-05 00:00:00
Tape Stop	VST3	Misc	0	1	2020-11-28 00:00:00
TapeDelay	VST2	Delay	1	0	2018-09-03 00:00:00
TDR Kotelnikov	VST3	Dynamics	1	0	2020-10-16 00:00:00
TDR Molotok	VST3	Dynamics	1	0	2020-10-16 00:00:00
TDR NOVA	VST3	EQ	1	0	2020-10-16 00:00:00
TDR VOS SlickEQ	VST3	EQ	1	0	2020-10-16 00:00:00
Temper	VST2	Dynamics	1	0	2019-04-11 00:00:00
Timeless 3	VST3	Delay	0	1	2021-08-08 00:00:00
Trance Gate	VST3	Gate	0	1	2020-11-28 00:00:00
Transient Shaper	VST3	Dynamics	0	1	2020-11-28 00:00:00
Valhalla FreqEcho	VST3	Delay	1	0	2020-07-11 00:00:00
Valhalla SpaceModula...	VST3	Filter	1	0	2020-07-11 00:00:00

Valhalla SuperMassive	VST3	Reverb	1	0	2020-07-11 00:00:00	
Valhalla UberMod	VST3	Chorus	0	1	2020-07-11 00:00:00	
Valhalla VintageVerb	VST3	Reverb	0	1	2020-07-11 00:00:00	
ValhallaDelay	VST3	Delay	0	1	2020-07-11 00:00:00	
ValhallaPlate	VST3	Reverb	0	1	2020-07-11 00:00:00	
ValhallaRoom	VST3	Reverb	0	1	2020-07-11 00:00:00	
ValhallaShimmer	VST3	Reverb	0	1	2020-07-11 00:00:00	
Vinyl	VST3	Distortion	1	1	2017-06-19 00:00:00	
VladG Limiter no.6	VST2	Dynamics	1	0	2020-10-16 00:00:00	
Vocal Doubler	VST3	Misc	1	1	2017-06-19 00:00:00	
VOLOCO	VST3	Pitch Correction	1	0	2019-07-14 00:00:00	

MySQL   localhost:3306 ssl   plugins   SQL > select * from fx;						
name	format	subtype	isFree	hasLicense	lastUpdate	
3-Band EQ	VST3	EQ	1	0	2020-11-28 00:00:00	
8 Bit Shaper	VST2	Distortion	1	0	2017-11-29 00:00:00	
ADHD Leveling Tool	VST2	Dynamics	1	0	2018-12-01 00:00:00	
Bitcrush	VST3	Distortion	0	1	2020-11-28 00:00:00	
Bitter Sweet V3	VST3	Dynamics	1	0	2019-04-11 00:00:00	
BX Cleansweep V2	VST3	Filter	1	1	2017-05-24 00:00:00	
BX_Rockrack V3 Player	VST3	Distortion	1	1	2017-05-24 00:00:00	
CamelCrusher	VST2	Distortion	1	0	2018-12-01 00:00:00	
CarveEQ	VST3	EQ	0	1	2020-11-28 00:00:00	
Chorus	VST3	Chorus	1	0	2020-11-28 00:00:00	
Comb Filter	VST3	Filter	0	1	2020-11-28 00:00:00	
Compressor	VST3	Dynamics	0	1	2020-11-28 00:00:00	
Convolver	VST3	Reverb	0	1	2020-11-28 00:00:00	
Correlometer	VST2	Monitoring	1	0	2017-09-22 00:00:00	
Delay	VST3	Delay	1	0	2020-11-28 00:00:00	
Delta Modulator	VST2	Distortion	1	0	2017-11-29 00:00:00	
Disperser	VST3	Filter	0	1	2020-11-28 00:00:00	
Distortion	VST3	Distortion	0	1	2020-11-28 00:00:00	
Dynamics	VST3	Dynamics	0	1	2020-11-28 00:00:00	
Ensemble	VST3	Chorus	0	1	2020-11-28 00:00:00	
Fatuator	VST3	Distortion	0	1	2020-11-28 00:00:00	
Filter	VST3	Filter	0	1	2020-11-28 00:00:00	
Flanger	VST3	Filter	0	1	2020-11-28 00:00:00	
Formant Filter	VST3	Filter	0	1	2020-11-28 00:00:00	
Fresh Air	VST3	Misc	1	0	2020-11-27 00:00:00	
Gain	VST3	Utility	1	0	2020-11-28 00:00:00	
Gate	VST3	Gate	0	1	2020-11-28 00:00:00	
GClip	VST2	Distortion	1	0	2020-12-27 00:00:00	
Glitch	VST2	Host	1	0	2019-04-11 00:00:00	
GSnap	VST2	Pitch Correction	1	0	2018-01-17 00:00:00	
Haas	VST3	Imaging	0	1	2020-11-28 00:00:00	
Krush	VST3	Distortion	1	0	2017-06-19 00:00:00	
Ladder Filter	VST3	Filter	0	1	2020-11-28 00:00:00	
Limiter	VST3	Dynamics	1	0	2020-11-28 00:00:00	
Loudness Meter II	VST3	Monitoring	1	0	2017-09-22 00:00:00	
Loudness Meter II Pro	VST3	Monitoring	0	1	2018-09-20 00:00:00	
MAutoPan	VST3	Modulation	1	0	2017-04-05 00:00:00	
MAutoPitch	VST3	Pitch Correction	1	0	2017-04-05 00:00:00	
MBandPass	VST3	Filter	1	0	2017-04-05 00:00:00	

MBitFun	VST3	Distortion	1	0	2017-04-05 00:00:00
MComb	VST3	Filter	1	0	2017-04-05 00:00:00
MCompressor	VST3	Dynamics	1	0	2017-04-05 00:00:00
MConvolutionEZ	VST3	Reverb	1	0	2017-04-05 00:00:00
Melodyne Essential	VST3	Pitch Correction	0	1	2020-08-21 00:00:00
MEqualizer	VST3	EQ	1	0	2017-04-05 00:00:00
MFlanger	VST3	Filter	1	0	2017-04-05 00:00:00
MFreqShifter	VST3	Misc	1	0	2017-04-05 00:00:00
MPhaser	VST3	Filter	1	0	2017-04-05 00:00:00
MRingModulator	VST3	Modulation	1	0	2017-04-05 00:00:00
MSaturator	VST3	Distortion	1	0	2017-04-05 00:00:00
MTremolo	VST3	Modulation	1	0	2017-04-05 00:00:00
Multipass	VST3	Host	0	1	2020-11-28 00:00:00
MVibrato	VST3	Pitch Shift	1	0	2017-04-05 00:00:00
MWaveFolder	VST3	Distortion	1	0	2017-04-05 00:00:00
Niveau Filter	VST3	Filter	1	1	2017-05-24 00:00:00
Nonlinear Filter	VST3	Filter	0	1	2020-11-28 00:00:00
OldSkoolVerb	VST2	Reverb	0	0	2017-09-22 00:00:00
OTT	VST2	Dynamics	1	0	2017-11-29 00:00:00
Ozone 9 Advanced	VST3	Mastering	0	1	2020-08-21 00:00:00
Ozone Imager II	VST3	Imaging	1	1	2017-06-19 00:00:00
PanCake II	VST2	Modulation	1	0	2020-12-27 00:00:00
Phase Distortion	VST3	Distortion	0	1	2020-11-28 00:00:00
Phaser	VST3	Filter	0	1	2020-11-28 00:00:00
Pitch Shifter	VST3	Pitch Shift	0	1	2020-11-21 00:00:00
Pro-C 2	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-DS	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-G	VST3	Gate	0	1	2021-08-08 00:00:00
Pro-L 2	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-MB	VST3	Dynamics	0	1	2020-11-25 00:00:00
Pro-Q 3	VST3	EQ	0	1	2019-04-30 00:00:00
Pro-R	VST3	Reverb	0	1	2020-11-25 00:00:00
Resonator	VST3	Misc	0	1	2020-11-28 00:00:00
Reverb	VST3	Reverb	0	1	2020-11-28 00:00:00
Reverser	VST3	Misc	0	1	2020-11-28 00:00:00
Ring Mod	VST3	Modulation	0	1	2020-11-28 00:00:00
Rough Rider 3	VST2	Dynamics	1	0	2018-10-08 00:00:00
Saturation Knob	VST3	Distortion	1	1	2017-06-19 00:00:00
Saturn 2	VST3	Distortion	0	1	2021-08-08 00:00:00
Slice EQ	VST3	EQ	0	1	2020-11-28 00:00:00
SnapHeap	VST3	Host	0	0	2020-11-28 00:00:00
SPAN	VST3	Monitoring	1	0	2017-09-22 00:00:00
Stereo	VST3	Imaging	1	0	2020-11-28 00:00:00

TAL Reverb 4	VST2	Reverb	1	0	2020-03-19 00:00:00
TAL-Vocoder	VST2	Misc	1	0	2017-04-05 00:00:00
Tape Stop	VST3	Misc	0	1	2020-11-28 00:00:00
TapeDelay	VST2	Delay	1	0	2018-09-03 00:00:00
TDR Kotelnikov	VST3	Dynamics	1	0	2020-10-16 00:00:00
TDR Molotok	VST3	Dynamics	1	0	2020-10-16 00:00:00
TDR NOVA	VST3	EQ	1	0	2020-10-16 00:00:00
TDR VOS SlickEQ	VST3	EQ	1	0	2020-10-16 00:00:00
Temper	VST2	Dynamics	1	0	2019-04-11 00:00:00
Timeless 3	VST3	Delay	0	1	2021-08-08 00:00:00
Trance Gate	VST3	Gate	0	1	2020-11-28 00:00:00
Transient Shaper	VST3	Dynamics	0	1	2020-11-28 00:00:00
Valhalla FreqEcho	VST3	Delay	1	0	2020-07-11 00:00:00
Valhalla SpaceModulator	VST3	Filter	1	0	2020-07-11 00:00:00
Valhalla SuperMassive	VST3	Reverb	1	0	2020-07-11 00:00:00
Valhalla UberMod	VST3	Chorus	0	1	2020-07-11 00:00:00
Valhalla VintageVerb	VST3	Reverb	0	1	2020-07-11 00:00:00
ValhallaDelay	VST3	Delay	0	1	2020-07-11 00:00:00
ValhallaPlate	VST3	Reverb	0	1	2020-07-11 00:00:00
ValhallaRoom	VST3	Reverb	0	1	2020-07-11 00:00:00
ValhallaShimmer	VST3	Reverb	0	1	2020-07-11 00:00:00
Vinyl	VST3	Distortion	1	1	2017-06-19 00:00:00
Vlado Limiter no.6	VST2	Dynamics	1	0	2020-10-16 00:00:00
Vocal Doubler	VST3	Misc	1	1	2017-06-19 00:00:00
VOLOCO	VST3	Pitch Correction	1	0	2019-07-14 00:00:00

107 rows in set (0.0220 sec)

MySQL localhost:3306 ssl SQL >

After delete:

```
mysql> delete from fx
-> where NOW() > DATE_ADD(fx.lastUpdate, INTERVAL 4 MONTH);
Query OK, 104 rows affected (0.01 sec)

mysql> select * from fx;
+-----+-----+-----+-----+-----+-----+
| name | format | subtype | isFree | hasLicense | lastUpdate |
+-----+-----+-----+-----+-----+-----+
| Pro-G | VST3 | Gate | 0 | 1 | 2021-08-08 00:00:00 |
| Saturn 2 | VST3 | Distortion | 0 | 1 | 2021-08-08 00:00:00 |
| Timeless 3 | VST3 | Delay | 0 | 1 | 2021-08-08 00:00:00 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Tables loaded!

name	format	subtype	isFree	hasLicense	lastUpdate
Pro-G	VST3	Gate	0	1	2021-08-08 00:00:00
Saturn 2	VST3	Distortion	0	1	2021-08-08 00:00:00
Timeless 3	VST3	Delay	0	1	2021-08-08 00:00:00

```
// insert 1: work with trigger 1
```

```
// for insert into downloaded table where user download new plugins at current time
insert into downloaded values(1, 'Haas', now());
```

Before insert:

Tables loaded!

uID	plugin	downloadDate
4	Bitcrush	2021-07-25 00:00:00
6	Convolver	2021-10-09 00:00:00
7	Distortion	2021-08-02 00:00:00
2	Gate	2021-10-02 00:00:00
4	Haas	2021-06-01 00:00:00
5	Haas	2021-08-09 00:00:00
6	Haas	2021-06-03 00:00:00
4	Multipass	2021-09-23 00:00:00
6	Multipass	2021-07-28 00:00:00
2	PhasePlant	2021-09-03 00:00:00
1	Pro-Q 3	2021-09-02 00:00:00
3	Reverb	2021-08-27 00:00:00

```
mysql> select * from downloaded;
+----+-----+-----+
| uID | plugin      | downloadDate   |
+----+-----+-----+
| 4  | Bitcrush    | 2021-07-25 00:00:00 |
| 6  | Convolver   | 2021-10-09 00:00:00 |
| 7  | Distortion   | 2021-08-02 00:00:00 |
| 2  | Gate         | 2021-10-02 00:00:00 |
| 4  | Haas         | 2021-06-01 00:00:00 |
| 5  | Haas         | 2021-08-09 00:00:00 |
| 6  | Haas         | 2021-06-03 00:00:00 |
| 4  | Multipass    | 2021-09-23 00:00:00 |
| 6  | Multipass    | 2021-07-28 00:00:00 |
| 2  | PhasePlant   | 2021-09-03 00:00:00 |
| 1  | Pro-Q 3     | 2021-09-02 00:00:00 |
| 3  | Reverb       | 2021-08-27 00:00:00 |
+----+-----+-----+
12 rows in set (0.00 sec)
```

After insert:

Tables loaded!

uID	plugin	downloadDate
4	Bitcrush	2021-07-25 00:00:00
6	Convolver	2021-10-09 00:00:00
7	Distortion	2021-08-02 00:00:00
2	Gate	2021-10-02 00:00:00
1	Haas	2021-12-04 22:49:10
4	Haas	2021-06-01 00:00:00
5	Haas	2021-08-09 00:00:00
6	Haas	2021-06-03 00:00:00
4	Multipass	2021-09-23 00:00:00
6	Multipass	2021-07-28 00:00:00
2	PhasePlant	2021-09-03 00:00:00
1	Pro-Q 3	2021-09-02 00:00:00
3	Reverb	2021-08-27 00:00:00

```
mysql> select * from downloaded;
+----+-----+-----+
| uID | plugin      | downloadDate          |
+----+-----+-----+
|   4 | Bitcrush    | 2021-07-25 00:00:00  |
|   6 | Convolver   | 2021-10-09 00:00:00  |
|   7 | Distortion   | 2021-08-02 00:00:00  |
|   2 | Gate         | 2021-10-02 00:00:00  |
|   1 | Haas         | 2021-12-04 22:49:10 |
|   4 | Haas         | 2021-06-01 00:00:00  |
|   5 | Haas         | 2021-08-09 00:00:00  |
|   6 | Haas         | 2021-06-03 00:00:00  |
|   4 | Multipass    | 2021-09-23 00:00:00  |
|   6 | Multipass    | 2021-07-28 00:00:00  |
|   2 | PhasePlant   | 2021-09-03 00:00:00  |
|   1 | Pro-Q 3      | 2021-09-02 00:00:00  |
|   3 | Reverb       | 2021-08-27 00:00:00  |
+----+-----+-----+
13 rows in set (0.00 sec)
```

// insert 2: insert a new user

Before insert:

Tables loaded!

uID	name	age	downLoaded
1	Kim	20	3
2	John	24	4
3	Amy	20	2
4	Lucas	28	6
5	Lio	19	2
6	Mike	30	6
7	Mia	21	3

```
mysql> select * from user;
+---+-----+-----+-----+
| uID | name  | age   | downLoaded |
+---+-----+-----+-----+
| 1  | Kim   | 20    | 3        |
| 2  | John  | 24    | 4        |
| 3  | Amy   | 20    | 2        |
| 4  | Lucas | 28    | 6        |
| 5  | Lio   | 19    | 2        |
| 6  | Mike  | 30    | 6        |
| 7  | Mia   | 21    | 3        |
+---+-----+-----+-----+
7 rows in set (0.00 sec)
```

After insert:

Tables loaded!

uID	name	age	downLoaded
1	Kim	20	3
2	John	24	4
3	Amy	20	2
4	Lucas	28	6
5	Lio	19	2
6	Mike	30	6
7	Mia	21	3
8	Anna	20	0

```

mysql> select * from user;
+----+-----+-----+-----+
| uID | name | age  | downLoaded |
+----+-----+-----+-----+
| 1   | Kim   | 20   | 3      |
| 2   | John  | 24   | 4      |
| 3   | Amy   | 20   | 2      |
| 4   | Lucas | 28   | 6      |
| 5   | Lio   | 19   | 2      |
| 6   | Mike  | 30   | 6      |
| 7   | Mia   | 21   | 3      |
| 8   | Anna  | 20   | 0      |
+----+-----+-----+-----+
8 rows in set (0.00 sec)

```

Note: automatic insertion of uID that is properly incremented with adding a new user

#### key constraint for User table

```
insert into user values(1, "jason", 20, 0);
```

```

java.sql.SQLIntegrityConstraintViolationException: Duplicate entry '1' for key 'user.PRIMARY'
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:117)
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122)
at com.mysql.cj.jdbc.StatementImpl.executeUpdateInternal(StatementImpl.java:1340)
at com.mysql.cj.jdbc.StatementImpl.executeLargeUpdate(StatementImpl.java:2089)
at com.mysql.cj.jdbc.StatementImpl.executeUpdate(StatementImpl.java:1251)
at application.Main.performSQLStatement(Main.java:99)

```

#### key constraint for Downloaded table

```
insert into downloaded values(2, "Gate", "2021-10-02");
```

```

Driver loaded!
java.sql.SQLIntegrityConstraintViolationException: Duplicate entry 'Delay' for key 'fx.PRIMARY'
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:117)
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122)
at com.mysql.cj.jdbc.StatementImpl.executeUpdateInternal(StatementImpl.java:1340)
at com.mysql.cj.jdbc.StatementImpl.executeLargeUpdate(StatementImpl.java:2089)
at com.mysql.cj.jdbc.StatementImpl.executeUpdate(StatementImpl.java:1251)

```

#### key constraint for plugins table

```
insert into plugins values("Chorus", "Slate Digital", "Generator");
```

```

Driver loaded!
java.sql.SQLIntegrityConstraintViolationException: Duplicate entry '2-Gate-2021-10-02 00:00:00' for key 'downloaded.PRIMARY'
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:117)
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122)
at com.mysql.cj.jdbc.StatementImpl.executeUpdateInternal(StatementImpl.java:1340)
at com.mysql.cj.jdbc.StatementImpl.executeLargeUpdate(StatementImpl.java:2089)
at com.mysql.cj.jdbc.StatementImpl.executeUpdate(StatementImpl.java:1251)
at application.Main.performSQLStatement(Main.java:99)

```

#### Foreign key constraint for Downloaded table update plugin

```
update downloaded set plugin = "Pro" where plugin = "Haas";
```

```

Driver loaded!
java.sql.SQLIntegrityConstraintViolationException: Cannot add or update a child row: a foreign key constraint fails ('plugins`.`downloaded', CONSTRAINT `downloaded_'
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:117)
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122)
at com.mysql.cj.jdbc.StatementImpl.executeUpdateInternal(StatementImpl.java:1340)
at com.mysql.cj.jdbc.StatementImpl.executeLargeUpdate(StatementImpl.java:2089)
at com.mysql.cj.jdbc.StatementImpl.executeUpdate(StatementImpl.java:1251)
at application.Main.performSQLStatement(Main.java:99)

```

Archive:

## Before archiving:

```
mysql> select * from archive;  
Empty set (0.00 sec)  
  
mysql> -
```

After archiving: