



北京理工大学珠海学院
BEIJING INSTITUTE OF TECHNOLOGY, ZHUHAI

数据库知识梳理

学 院： 数理与土木工程学院
课 程： 数据库技术基础
班 级： 数据科学与大数据技术 2 班
姓 名： 肖宇涵
学 号： 211205101786

中国·珠海

2022 年 12 月 30 日

目 录

1	MySQL 中的正则表达式	1
2	MySQL 中的新增、删除、修改	6
2.1	常见新增操作汇总:	6
2.1.1	全字段单条数据插入	6
2.1.2	指定字段单条插入	7
2.1.3	多条数据插入	7
2.1.4	替换插入	8
2.1.5	表中数据复制	8
2.2	常见删除操作汇总	9
2.2.1	根据条件删除:	9
2.2.2	全部删除 (表清空, 包含自增计数器重置):	10
2.3	常见的修改操作	11
3	MySQL 中的索引	12
3.1	索引的优缺点	12
3.1.1	优点	12
3.1.2	缺点	12
3.2	索引的基本语法	13
3.2.1	创建	13
3.2.2	删除	13
3.2.3	查看	13
3.3	索引的分类	13
3.3.1	普通索引	13
3.3.2	唯一索引	14
3.3.3	主键索引	15
3.3.4	单列索引	16
3.3.5	组合索引	16
3.3.6	全文索引	16
4	MySQL 中的事务	17

1 MySQL 中的正则表达式

MySQL 正则表达式通常是在检索数据库记录的时候，根据指定的匹配模式匹配记录中符合要求的特殊字符串

MySQL 的正则表达式使用 REGEXP 这个关键字来指定正则表达式的匹配模式，REGEXP 操作符所支持的匹配模式如表所示

匹配	描述
^	匹配文本的开始字符
\$	匹配文本的结束字符
.	匹配任何单个字符
*	匹配零个或多个在它前面的字符
%	匹配所有
+	匹配前面的字符 1 次或多次
?	零个或一个
字符串	匹配包含指定的字符串
p1 p2	匹配 p1 或 p2
[...]	匹配字符集中的任意一个字符
[^...]	匹配不在括号中的任何字符
{n}	匹配前面的字符串 n 次
{n,m}	匹配前面的字符串至少 n 次，至多 m 次

用美妆销售数据分析中的数据，查询顾客的统计时间开头为 2016 的数据

```

1 select update_time ,id
2     from 肖宇涵.beauty_data
3     where update_time regexp '^2016'          ##查询以2016开头的学生信息

```

输出结果如下：

select update time ,id from 肖宇涵.beauty c

	update_time	id
185	2016-11-8	A527081433316
186	2016-11-8	A527082291331
187	2016-11-8	A527089635818
188	2016-11-8	A527129766073
189	2016-11-8	A527621891313
190	2016-11-8	A527778412235
191	2016-11-8	A528381038628
192	2016-11-8	A528596459318
193	2016-11-8	A529658394685
194	2016-11-8	A529658846767
195	2016-11-8	A530228778402
196	2016-11-8	A531970304102
197	2016-11-8	A532558830570
198	2016-11-8	A533900013929
199	2016-11-8	A534995666285
200	2016-11-8	A535435133234
201	2016-11-8	A535609137028
202	2016-11-8	A536031278248

用美妆销售数据分析中的数据，查询顾客的统计时间结尾为 8 的数据

```

1 select update_time ,id
2     from 肖宇涵.beauty_data
3     where update_time regexp '8$'      ##查询以8结尾的的信息

```

输出结果如下：

select update_time ,id from 肖宇涵.beauty_data

	update_time	id
185	2016-11-8	A527081433316
186	2016-11-8	A527082291331
187	2016-11-8	A527089635818
188	2016-11-8	A527129766073
189	2016-11-8	A527621891313
190	2016-11-8	A527778412235
191	2016-11-8	A528381038628
192	2016-11-8	A528596459318
193	2016-11-8	A529658394685
194	2016-11-8	A529658846767
195	2016-11-8	A530228778402
196	2016-11-8	A531970304102
197	2016-11-8	A532558830570
198	2016-11-8	A533900013929
199	2016-11-8	A534995666285
200	2016-11-8	A535435133234
201	2016-11-8	A535609137028
202	2016-11-8	A536031278248

用美妆销售数据分析中的数据，查询顾客的统计时间为 14 号的数据

```

1 select update_time ,id
2     from 肖宇涵.beauty_data
3     where update_time regexp '14'      ##查询中统计时间有14的信息

```

输出结果如下：

	update_time	id
1	2016-11-14	A18164178225
2	2016-11-14	A18177105952
3	2016-11-14	A18177226992
4	2016-11-14	A18178033846
5	2016-11-14	A18178045259
6	2016-11-14	A18178129035
7	2016-11-14	A18178206572
8	2016-11-14	A18190290933
9	2016-11-14	A18250630014
10	2016-11-14	A18422797881
11	2016-11-14	A18919906680
12	2016-11-14	A18934397934
13	2016-11-14	A18934597093
14	2016-11-14	A18980833277
15	2016-11-14	A18981009694
16	2016-11-14	A18997821811
17	2016-11-14	A18997825320
18	2016-11-14	A19008350212

用美妆销售数据分析中的数据，查询顾客的统计时间为 2016-11-13 或者 2016-11-14 号的数据

```

1 select update_time ,id
2     from 肖宇涵.beauty_data
3     where update_time regexp '2016-11-13|2016-11-14';
4     ##查询顾客的统计时间为2016-11-13或者2016-11-14的信息

```

输出结果如下：

ABC update_time	ABC id
2016-11-14	A538605631655
2016-11-14	A539341851212
2016-11-14	A539343210732
2016-11-14	A539343870010
2016-11-14	A539344100318
2016-11-14	A539793941051
2016-11-14	A539970643513
2016-11-14	A540215001957
2016-11-13	A18164178225
2016-11-13	A18177105952
2016-11-13	A18177226992
2016-11-13	A18178033846
2016-11-13	A18178045259
2016-11-13	A18178129035
2016-11-13	A18178206572
2016-11-13	A18190290933
2016-11-13	A18250630014
2016-11-13	A18422797881

用美妆销售数据分析中的数据，查询产品名称中有 CHANDO，而自然堂可有可无的信息

```

1 select update_time ,id, title
2     from 肖宇涵.beauty_data bd
3     where title regexp 'CHANDO/自然堂*'; ##查询产品名称中有CHANDO，自
      然堂可有可无的信息

```

输出结果如下:

	update_time	id	title
1	2016-11-14	A18164178225	CHANDO/自然堂 雪域精粹纯粹滋润霜50g 补水保湿 滋润水润面霜
2	2016-11-14	A18177105952	CHANDO/自然堂凝时鲜颜肌活乳液120ML 淡化细纹补水滋润专柜正品
3	2016-11-14	A18177226992	CHANDO/自然堂活泉保湿修护精华水 (滋润型)135ml 补水控油爽肤水
4	2016-11-14	A18178033846	CHANDO/自然堂 男士劲爽控油洁面膏 100g 深层清洁 男士洗面奶
5	2016-11-14	A18178045259	CHANDO/自然堂雪域精粹纯粹滋润霜 (清爽型) 50g补水保湿滋润霜
6	2016-11-14	A18178206572	CHANDO/自然堂雪润皙白晶采霜 (滋润型) 50g 滋润修护面霜 正品
7	2016-11-14	A18250630014	CHANDO/自然堂 凝时鲜颜肌活修护精华液35ml 淡化细纹护肤精华
8	2016-11-14	A18422797881	CHANDO/自然堂活泉保湿精华套装 保湿补水滋润控油面部护理套装
9	2016-11-14	A18934397934	CHANDO/自然堂水润保湿霜50g 补水保湿锁水细腻清爽防干燥正品
10	2016-11-14	A18934597093	CHANDO/自然堂水润长效保湿霜50g 深层补水持久锁水细腻清爽滋润
11	2016-11-14	A18980833277	CHANDO/自然堂男士面部护肤套装 洁面洗面奶爽肤水面霜
12	2016-11-14	A18981009694	CHANDO/自然堂雪润深澈精华面膜贴5片 补水面膜护肤品专柜
13	2016-11-14	A18997821811	CHANDO/自然堂娇嫩嫩白保湿霜50g 保湿补水滋润清爽保湿霜
14	2016-11-14	A18997825320	CHANDO/自然堂娇颜 保湿乳100ML 持久保湿水嫩肌肤补水 乳液
15	2016-11-14	A19008350212	CHANDO/自然堂雪润皙白水乳套装 爽肤水乳液面膜 亮肤补水套装
16	2016-11-14	A19009618209	CHANDO/自然堂凝时鲜颜套装水乳/霜套装 补水保湿滋润淡化细纹
17	2016-11-14	A19371494116	CHANDO/自然堂雪域精粹冰肌精华套组 滋润护肤 精华爽肤水BB套装
18	2016-11-14	A19742978078	CHANDO/自然堂水润保湿面膜5片装 滋润补水保湿面膜护肤品专柜 正品

用美妆销售数据分析中的数据, 查询产品名称中含有保湿, 水至少出现一次的信息

```

1 select update_time ,id, title
2     from 肖宇涵.beauty_data bd
3     where title regexp '保湿水+'; ##查询产品名称中含有保湿, 水至少出现一
    次的信息

```

输出结果如下:

	update_time	id	title
173	2016-11-9	A122296	专柜正品相宜本草芯净自然净肤保湿水清爽补水化妆品女秋控油护肤
174	2016-11-9	A180313	【真5折狂欢购】相宜本草黑茶男士护肤套装洗面奶乳液保湿水控油
175	2016-11-9	A195609	【4.7折抢先预售】相宜本草百合高保湿水面膜套装补水保湿滋润
176	2016-11-9	A220277	【真5折狂欢购】相宜本草百合高保湿水凝霜补水保湿修护面霜女
177	2016-11-9	A355555	【真5折狂欢购】相宜本草红石榴鲜活亮白嫩肤精华水保湿水2瓶
178	2016-11-9	A520305	【领券满199减100】相宜本草睡莲莹泽保湿水女补水爽肤护肤品
179	2016-11-9	A537490	【新品上市】相宜本草野果果保湿水份乳液润泽滋润补水锁水
180	2016-11-9	A537524	正品相宜本草芯净自然保湿喷雾女补水滋养清爽保湿水
181	2016-11-9	A538758	【预售买就送水乳中样】相宜本草金缕梅补水保湿水乳祛痘凝胶套装
182	2016-11-8	A122296	专柜正品相宜本草芯净自然净肤保湿水清爽补水化妆品女秋控油护肤
183	2016-11-8	A180313	【真5折狂欢购】相宜本草黑茶男士护肤套装洗面奶乳液保湿水控油
184	2016-11-8	A195609	【4.7折抢先预售】相宜本草百合高保湿水面膜套装补水保湿滋润
185	2016-11-8	A220277	【真5折狂欢购】相宜本草百合高保湿水凝霜补水保湿修护面霜女
186	2016-11-8	A355555	【正装买一送一】相宜本草红石榴鲜活亮白嫩肤精华水保湿水2瓶
187	2016-11-8	A520305	【领券满199减100】相宜本草睡莲莹泽保湿水女补水爽肤护肤品
188	2016-11-8	A537490	【新品上市】相宜本草野果果保湿水份乳液润泽滋润补水锁水
189	2016-11-8	A537524	正品相宜本草芯净自然保湿喷雾女补水滋养清爽保湿水
190	2016-11-8	A538758	【预售买就送水乳中样】相宜本草金缕梅补水保湿水乳祛痘凝胶套装

用美妆销售数据分析中的数据, 查询产品名称中以雅-雪开头的信息

```

1 select update_time ,id, title
2     from 肖宇涵.beauty_data bd
3     where title regexp '^[雅-雪]'; ##查询产品名称以雅-雪开头的信息

```

输出结果如下:

	update_time	id	title
1346	2016-11-5	A528060	雅诗兰黛眼霜 多效智妍滋养眼霜15ml 提拉紧致 深透润泽 补水保湿
1347	2016-11-5	A528116	雅诗兰黛眼霜 弹性紧实柔肤眼霜15ml 提拉紧致淡化细纹补水
1348	2016-11-5	A529418	雅诗兰黛粉底液 持妆无瑕液体粉底 遮瑕持久保湿不脱妆
1349	2016-11-5	A533036	雅诗兰黛BB霜 晶透沁白双效修颜霜30ml 防护隔离
1350	2016-11-5	A539046	雅诗兰黛多效智妍精华霜75ml+智妍眼霜15ml淡化细纹
1351	2016-11-5	A539048	雅诗兰黛 红石榴鲜亮焕采泡沫洁面乳 60ml
1352	2016-11-5	A539172	雅诗兰黛 鲜活亮采果萃水 (滋润型) 100ml
1353	2016-11-5	A541125	雅诗兰黛 白金级蕴能黑钻焕肤面膜 50ml
1354	2016-11-14	A526144	雪花秀润珪肌本赋活精华露50ml 精华液 补水保湿 紧致弹力 精华
1355	2016-11-14	A526144	雪花秀滋晶雪肤美白修护眼霜15ml 补水保湿 淡化眼袋 修护眼霜
1356	2016-11-14	A526147	雪花秀臻秀修护活颜水125ml 保湿水补水保湿柔肤水 紧致爽肤水
1357	2016-11-14	A526147	雪花秀臻秀修护眼霜25ml 眼部精华霜 淡细纹改善黑眼圈紧致
1358	2016-11-14	A526148	雪花秀顺行柔和活颜油200ml 卸妆油正品脸部眼唇 温和卸妆 不油腻
1359	2016-11-14	A526148	雪花秀玉璨净爽面膜150ml玉容撕拉面膜
1360	2016-11-14	A526149	雪花秀清润舒缓面膜80ml 睡眠面膜深层补水保湿夜间修护收缩毛孔
1361	2016-11-14	A526179	雪花秀润致焕活肌底精华露90ml 润燥精华 保湿补水 润泽滋养
1362	2016-11-14	A526184	雪花秀臻秀修护活颜霜60ml 紧致补水保湿淡化细纹 护肤面霜 滋润
1363	2016-11-14	A526184	雪花秀与润修复睡眠面膜120ml 韩国夜间面膜

用美妆销售数据分析中的数据，不是 2016-11-13 的信息

```

1 select update_time ,id, title
2     from 肖宇涵.beauty_data
3     where update_time regexp '[^2016\\-11\\-13]'
4     ##查询时间不是2016-11-13的信息

```

输出结果如下:

	update_time	id	title
1	2016-11-14	A18164178225	CHANDO/自然堂 雪域精粹纯粹滋润霜50g 补水保湿 滋润水润面霜
2	2016-11-14	A18177105952	CHANDO/自然堂凝时鲜颜肌活乳液120ML 淡化细纹补水滋润专柜正品
3	2016-11-14	A18177226992	CHANDO/自然堂活泉保湿修护精华水 (滋润型)135ml 补水控油爽肤水
4	2016-11-14	A18178033846	CHANDO/自然堂 男士劲爽控油洁面膏 100g 深层清洁 男士洗面奶
5	2016-11-14	A18178045259	CHANDO/自然堂雪域精粹纯粹滋润霜 (清爽型) 50g补水保湿滋润霜
6	2016-11-14	A18178129035	自然堂 雪域纯粹滋润洗颜霜 110g 补水保湿 洗面奶女 深层清洁
7	2016-11-14	A18178206572	CHANDO/自然堂雪润皙白晶采霜 (滋润型) 50g 滋润修护面霜 正品
8	2016-11-14	A18190290933	自然堂 活泉深层净化控油凝露60g 控油补水保湿滋润 活泉精华正品
9	2016-11-14	A18250630014	CHANDO/自然堂 凝时鲜颜肌活修护精华液35ml 淡化细纹护肤精华
10	2016-11-14	A18422797881	CHANDO/自然堂活泉保湿精华套装 保湿补水滋润控油面部护理套装
11	2016-11-14	A18919906680	自然堂 水润保湿柔肤水 135ml 补水保湿清透锁水 专柜正品
12	2016-11-14	A18934397934	CHANDO/自然堂水润保湿霜50g 补水保湿锁水细腻清爽防干燥正品
13	2016-11-14	A18934597093	CHANDO/自然堂水润长效保湿霜50g 深层补水持久锁水细腻清爽滋润
14	2016-11-14	A18980833277	CHANDO/自然堂男士面部护肤套装 洁面洗面奶爽肤水面霜
15	2016-11-14	A18981009694	CHANDO/自然堂雪润深澈精华面膜贴5片 补水面膜护肤品专柜
16	2016-11-14	A18997821811	CHANDO/自然堂娇颜嫩白保湿霜50g 保湿补水滋润清保湿霜
17	2016-11-14	A18997825320	CHANDO/自然堂娇颜 保湿乳100ML 持久保湿水嫩肌肤补水 乳液
18	2016-11-14	A19008350212	CHANDO/自然堂雪润皙白水乳套装 爽肤水乳洁面乳 享肤补水套装

用美妆销售数据分析中的数据，查询产品名称不以“C雪自”开头的产品信息

```

1 select update_time ,id, title
2     from 肖宇涵.beauty_data
3     where title regexp '[^C雪自]';
4     ##查询学生名字不以“C雪自”开头的学生信息

```

输出结果如下:

	update_time	abc id	abc title
1	2016-11-14	A24573628465	专柜正品自然堂 活泉加倍保湿霜50g 补水锁水减少细纹滋润肌肤
2	2016-11-13	A24573628465	专柜正品自然堂 活泉加倍保湿霜50g 补水锁水减少细纹滋润肌肤
3	2016-11-10	A24573628465	专柜正品自然堂 活泉加倍保湿霜50g 补水锁水减少细纹滋润肌肤
4	2016-11-9	A24573628465	专柜正品自然堂 活泉加倍保湿霜50g 补水锁水减少细纹滋润肌肤
5	2016-11-8	A24573628465	专柜正品自然堂 活泉加倍保湿霜50g 补水锁水减少细纹滋润肌肤
6	2016-11-5	A24573628465	专柜正品自然堂 活泉加倍保湿霜50g 补水锁水减少细纹滋润肌肤
7	2016-11-14	A41238850925	资生堂心机彩妆臻采腮红 2g 官方正品
8	2016-11-14	A41276592245	资生堂洗面奶 男士洗面膏125ml 可作剃须泡沫
9	2016-11-14	A41290443958	shiseido资生堂 盼丽风姿金采丰润柔肤液150ml 水润紧致柔滑肌肤
10	2016-11-14	A41292111153	资生堂盼丽风姿抗皱夜霜50ml
11	2016-11-14	A41309234345	shiseido资生堂 盼丽风姿抗皱夜乳 75mL补水 保湿 官方正品
12	2016-11-14	A41310047163	资生堂 全球彩妆丝柔滑眼影组3g 妆效持久不晕染 眼影
13	2016-11-14	A41324454533	资生堂 全球彩妆瓷光蜜釉口红6ml 自然持久妆容 资生堂口红 唇膏
14	2016-11-14	A41336958216	shiseido资生堂心机彩妆臻采透无瑕粉底液 保湿控油逆光隔离
15	2016-11-14	A41338910703	shiseido资生堂 悦薇美颈精华露 75g滋润 保湿 官方正品
16	2016-11-14	A41342851949	Shiseido/资生堂新溪美肌亮润眼霜15ml
17	2016-11-14	A41348433513	资生堂眼膜 悦薇质纯防皱眼膜贴1对x12袋
18	2016-11-14	A41353789565	资生堂 全球彩妆至萃浓密睫毛膏 (精致型) 8ml 浓密睫毛 睫毛膏

2 MySQL 中的新增、删除、修改

用美妆销售数据分析中的数据，查询产品名称不以“C 雪自”开头的产品信息

2.1 常见新增操作汇总：

常见新增操作

MySQL 数据库中常使用 insert into 关键字添加数据，在使用 insert into 过程中，最常见的就是：

```
1 insert into tb_name(字段1,字段2, 字段3... )
2 values('data1', 'data2', 'data3' ...);
```

但除了上述的操作之外，还有很多种用法

2.1.1 全字段单条数据插入

全字段就是一个表中所有的字段，包括自增的主键（该字段值可写为 null）

```
1 insert into tb_name
2 values('data1', 'data2', 'data3' ...)
3 -- values();中必须包含该表中的所有字段
```

以课上习题表 user-profile 为例，插入一行数据

```
1 insert into 肖字涵.user_profile
2 values(NULL, 'female', '20', '北京理工大学珠海学院', '4.0', '66');
```

输出结果如下：

	123 DEVICE_ID ▼	ABC GENDER ▼	123 AGE ▼	ABC UNIVERSITY ▼	123 GPA ▼	123 ACTIVE_DAYS_WITHIN_30 ▼
1	2,138	male	21	北京大学	3.4	7
2	3,214	male	[NULL]	复旦大学	4	15
3	6,543	female	20	北京大学	3.2	12
4	2,315	female	23	浙江大学	3.6	5
5	5,432	male	25	山东大学	3.8	20
6	2,131	male	28	山东大学	3.3	15
7	4,321	male	28	复旦大学	3.6	9
8	[NULL]	female	20	北京理工大学珠海学院	4	66

2.1.2 指定字段单条插入

```
1 insert into tb_name(字段1,字段2, 字段3... )
2 values('data1', 'data2', 'data3' ...);
```

以课上习题表 user-profile 为例，插入一行数据

```
1 insert into 肖宇涵.user_profile (DEVICE_ID,GENDER,AGE,UNIVERSITY,GPA,
  ACTIVE_DAYS_WITHIN_30)
2 values(NULL, 'male', '20', '北京理工大学珠海学院', '4.0', '99');
```

输出结果如下：

	123 DEVICE_ID ▼	ABC GENDER ▼	123 AGE ▼	ABC UNIVERSITY ▼	123 GPA ▼	123 ACTIVE_DAYS_WITHIN_30 ▼
1	2,138	male	21	北京大学	3.4	7
2	3,214	male	[NULL]	复旦大学	4	15
3	6,543	female	20	北京大学	3.2	12
4	2,315	female	23	浙江大学	3.6	5
5	5,432	male	25	山东大学	3.8	20
6	2,131	male	28	山东大学	3.3	15
7	4,321	male	28	复旦大学	3.6	9
8	[NULL]	male	20	北京理工大学珠海学院	4	99

2.1.3 多条数据插入

多条数据插入又分为全字段和指定字段，和上述一样，下来只介绍全字段的多条数据插入，注意的是，多条插入中，values 只写一个，括号后用逗号隔开

```
1 insert into tb_name
2 values('data1', 'data2', 'data3', 'data4' ...),
3 ('data1', 'data2', 'data3', 'data4' ...),
4 ('data1', 'data2', 'data3', 'data4' ...),
5 ('data1', 'data2', 'data3', 'data4' ...),
6 ('data1', 'data2', 'data3', 'data4' ...);
```

以课上习题表 user-profile 为例，插入多条数据

```
1 insert into 肖宇涵.user_profile
2 values(NULL, 'female', '20', '北京理工大学珠海学院', '4.0', '33'),
3 (NULL, 'female', '20', '北京理工大学珠海学院', '4.0', '55'),
4 (NULL, 'male', '20', '北京理工大学珠海学院', '4.0', '99');
```

输出结果如下:

	123 DEVICE_ID ▼	abc GENDER ▼	123 AGE ▼	abc UNIVERSITY ▼	123 GPA ▼	123 ACTIVE_DAYS_WITHIN_30 ▼
1	2,138	male	21	北京大学	3.4	7
2	3,214	male	[NULL]	复旦大学	4	15
3	6,543	female	20	北京大学	3.2	12
4	2,315	female	23	浙江大学	3.6	5
5	5,432	male	25	山东大学	3.8	20
6	2,131	male	28	山东大学	3.3	15
7	4,321	male	28	复旦大学	3.6	9
8	[NULL]	male	20	北京理工大学珠海学院	4	99
9	[NULL]	female	20	北京理工大学珠海学院	4	33
10	[NULL]	female	20	北京理工大学珠海学院	4	55
11	[NULL]	male	20	北京理工大学珠海学院	4	99

2.1.4 替换插入

MySQL 除了使用 insert into 插入数据之外, 还有 replace into, 顾名思义, 如果表中存在相同主键的数据, 则覆盖它。例如, 学生表中有学号为 '2120802', 姓名为 '张三' 的一条数据, 该学生已经毕业, 此时, 李四同学使用该学号可直接使用 replace into 替换掉张三的数据, 其用法和 insert into 一样, 分为全字段, 指定字段, 多条和单条, 以下只演示指定字段单条数据替换插入用法:

```
1 replace into tb_name(字段1,字段2, 字段3... )
2 values('data1', 'data2', 'data3' ...);
```

以课上习题表 user-profile 为例, 替换插入数据

```
1 replace into 肖宇涵.user_profile (DEVICE_ID,GENDER,AGE,UNIVERSITY,GPA,
  ACTIVE_DAYS_WITHIN_30)
2 values('211205', 'female', '20','北京理工大学珠海学院','4.0', '33');
```

	123 DEVICE_ID ▼	abc GENDER ▼	123 AGE ▼	abc UNIVERSITY ▼	123 GPA ▼	123 ACTIVE_DAYS_WITHIN_30 ▼
1	2,138	male	21	北京大学	3.4	7
2	3,214	male	[NULL]	复旦大学	4	15
3	6,543	female	20	北京大学	3.2	12
4	2,315	female	23	浙江大学	3.6	5
5	5,432	male	25	山东大学	3.8	20
6	2,131	male	28	山东大学	3.3	15
7	4,321	male	28	复旦大学	3.6	9
8	[NULL]	male	20	北京理工大学珠海学院	4	99
9	[NULL]	female	20	北京理工大学珠海学院	4	33
10	[NULL]	female	20	北京理工大学珠海学院	4	55
11	[NULL]	male	20	北京理工大学珠海学院	4	99
12	[NULL]	male	20	北京理工大学珠海学院	4	99

2.1.5 表中数据复制

如果把一个表中符合条件的数据复制到另一张表中, 可以使用该方法, 该语法中没有 values 关键字, select 语句把符合条件的数据从已知表中查询出来作为插入目标表中的数据

语法格式如下:

```

1 insert into tb_name
2 select 字段1, 字段2, 字段3 ...
3 from tb_name2
4 where 条件表达式

```

先建立一张与 user-profile 表结构相同的表 user-profile1，建表语句如下

```

1 CREATE TABLE 'user_profile1' (
2 'DEVICE_ID' int DEFAULT NULL,
3 'GENDER' varchar(10) DEFAULT NULL,
4 'AGE' tinyint DEFAULT NULL,
5 'UNIVERSITY' varchar(20) DEFAULT NULL,
6 'GPA' float (3,2) DEFAULT NULL,
7 'ACTIVE_DAYS_WITHIN_30' tinyint DEFAULT null)

```

将 user-profile 复制到 user-profile1 中，代码如下

```

1 insert into user_profile1
2 select DEVICE_ID,GENDER,AGE,UNIVERSITY,GPA,ACTIVE_DAYS_WITHIN_30
3 from 肖宇涵.user_profile

```

The screenshot shows a SQL query result in a database client. The query is `select * from 肖宇涵.user_profile1`. The result is a table with 7 columns: DEVICE_ID, GENDER, AGE, UNIVERSITY, GPA, and ACTIVE_DAYS_WITHIN_30. The data is as follows:

	DEVICE_ID	GENDER	AGE	UNIVERSITY	GPA	ACTIVE_DAYS_WITHIN_30
1	2,138	male	21	北京大学	3.4	7
2	3,214	male	[NULL]	复旦大学	4	15
3	6,543	female	20	北京大学	3.2	12
4	2,315	female	23	浙江大学	3.6	5
5	5,432	male	25	山东大学	3.8	20
6	2,131	male	28	山东大学	3.3	15
7	4,321	male	28	复旦大学	3.6	9
8	[NULL]	male	20	北京理工大学珠海学院	4	99
9	[NULL]	female	20	北京理工大学珠海学院	4	33
10	[NULL]	female	20	北京理工大学珠海学院	4	55
11	[NULL]	male	20	北京理工大学珠海学院	4	99
12	[NULL]	male	20	北京理工大学珠海学院	4	99

2.2 常见删除操作汇总

2.2.1 根据条件删除：

```

1 DELETE FROM tb_name [WHERE options]
2 [ [ ORDER BY fields ] LIMIT n ]

```

limit 的用法有两种：

带一个参数 `limit n`：表示罗列查询后的前 `n` 项数据
带两个参数 `limit m, n`：表示从第 `m` 行数据开始，罗列出 `n` 行数据

这里我们还是以 user-profile 为例，按条件删除 AGE>25 的学生跟 GPA=4 的学生

```
1 insert into 肖宇涵.user_profile
2 values(NULL, 'female', '20', '北京理工大学珠海学院', '4.0', '66');
```

代码展示如下：

```
1 delete
2 from 肖宇涵.user_profile1
3 where AGE>25
4 or GPA=4
```

输出结果如下：

Name	Value
Updated Rows	8
Query	delete
	from 肖宇涵.user_profile1
	where AGE>25
	or GPA=4
Finish time	Sat Dec 03 13:44:11 CST 2022

123	DEVICE_ID	ABC	GENDER	123	AGE	ABC	UNIVERSITY	123	GPA	123	ACTIVE_DAYS_WITHIN_30
1	2,138		male		21		北京大学		3.4		7
2	6,543		female		20		北京大学		3.2		12
3	2,315		female		23		浙江大学		3.6		5
4	5,432		male		25		山东大学		3.8		20

2.2.2 全部删除（表清空，包含自增计数器重置）：

```
1 TRUNCATE tb_name
```

示例，代码如下：

```
1 TRUNCATE user_profile1
```

输出结果如下：

Name	Value
Updated Rows	0
Query	TRUNCATE user_profile1
Finish time	Sat Dec 03 03:42:11 CST 2022

[illegible]

2.3 常见的修改操作

MySQL 数据库中使用 update 关键字来修改表中数据更新操作的常用语法格式如下:

```
1  设置为新值： UPDATE table_name SET
2      column_name=new_value
3      [, column_name2=new_value2]
4      [WHERE column_name3=value3]
5  根据已有值替换： UPDATE table_name SET
6      key1=replace(key1, '查找内容', '替换成内容')
7      [WHERE column_name3=value3]
```

这里我们还是以 `user-profile` 为例，按条件将 `DEVICE-ID` 的数值修改为 1
示例代码如下

```
1 UPDATE user_profile1
2     set DEVICE_ID= 1
3 ORDER BY DEVICE_ID DESC;
```

输出结果如下:

	123 DEVICE_ID 🚩	abc GENDER 🚩	123 AGE 🚩	abc UNIVERSITY 🚩	123 GPA 🚩	123 ACTIVE_DAYS_WITHIN_30 🚩
1	1	male	21	北京大学	3.4	7
2	1	male	[NULL]	复旦大学	4	15
3	1	female	20	北京大学	3.2	12
4	1	female	23	浙江大学	3.6	5
5	1	male	25	山东大学	3.8	20
6	1	male	28	山东大学	3.3	15
7	1	male	28	复旦大学	3.6	9
8	1	male	20	北京理工大学珠海学院	4	99
9	1	female	20	北京理工大学珠海学院	4	33
10	1	female	20	北京理工大学珠海学院	4	55
11	1	male	20	北京理工大学珠海学院	4	99
12	1	male	20	北京理工大学珠海学院	4	99

示例代码如下

```
1 UPDATE user_profile1
2     set DEVICE_ID= DEVICE_ID+1
3 ORDER BY DEVICE ID DESC;
```

输出结果如下:

	123 DEVICE_ID ▼	abc GENDER ▼	123 AGE ▼	abc UNIVERSITY ▼	123 GPA ▼	123 ACTIVE_DAYS_WITHIN_30 ▼
1	2	male	21	北京大学	3.4	7
2	2	male	[NULL]	复旦大学	4	15
3	2	female	20	北京大学	3.2	12
4	2	female	23	浙江大学	3.6	5
5	2	male	25	山东大学	3.8	20
6	2	male	28	山东大学	3.3	15
7	2	male	28	复旦大学	3.6	9
8	2	male	20	北京理工大学珠海学院	4	99
9	2	female	20	北京理工大学珠海学院	4	33
10	2	female	20	北京理工大学珠海学院	4	55
11	2	male	20	北京理工大学珠海学院	4	99
12	2	male	20	北京理工大学珠海学院	4	99

3 MySQL 中的索引

MySQL 官方对索引的定义为：索引是帮助 MySQL 高效获取数据的数据结构。索引的本质是数据结构。我们自己可以理解为：

数据本身之外，数据库还维护着一个满足特定查找算法的数据结构，这些数据结构以某种方式指向数据，这样就可以在这些数据结构的基础上实现高级查找算法，这种数据结构就是索引。

我们平常所说的索引，如果没有特别指明，都是指 B 树（多路搜索树，并不一定是二叉的）结构组织的索引。其中聚集索引，次要索引，覆盖索引，复合索引，前缀索引，唯一索引默认都是使用 B+ 树索引，统称索引。当然，除了 B+ 树这种类型的索引之外，还有哈希索引（hash index）等

3.1 索引的优缺点

3.1.1 优点

1. 类似大学图书馆建书目索引，提高数据检索的效率，降低数据库的 IO 成本
2. 通过索引列对数据进行排序，降低数据排序的成本，降低了 CPU 的消耗

3.1.2 缺点

1. 实际上索引也是一张表，该表保存了主键与索引字段，并指向实体表的记录，所以索引列也是要占用空间的
2. 虽然索引大大提高了查询速度，同时却会降低更新表的速度，INSERT 如对表进行、UPDATE 和 IDELETE，因为更新表时，MySQL 不仅要保存数据，还要保存一下索引文件每次更新添加了索引列的字段，都会调整因为更新所带来的键值变化后的索引信息
3. 索引只是提高效率的一个因素，如果你的 MYSQL 有大数据量的表，就需要花时间研究建立最优秀的索引，或优化查询语句

3.2 索引的基本语法

3.2.1 创建

其中加 unique 表示创建的是唯一索引

```
1 #CREATE[UNIQUE] INDEX 索引名 ON 表名(字段名(length))
2 CREATE[UNIQUE] INDEX index_name ON mytable(columnname(length))
3 #ALTER table 表名 ADD [UNIQUE INDEX [索引名] ON(字段名(length))
4 ALTER table mytable ADD [UNIQUE INDEX [index_name] ON(columnname(length))
```

如果是 CHAR, VARCHAR 类型, length 可以小于字段实际长度; 如果是 BLOB 和 TEXT 类型, 必须指定 length 使用 alter 命令

```
1 有四种方式来添加数据表的索引:
2 ALTER TABLE tbl_name ADD PRIMARY KEY(column_list)该语句添加一个主键,这意味着索引值必须是唯一的,且不能为NULL
3 ALTER TABLE tbl_name ADD UNIQUE index_name(column_list)这条语句创建索引的值必须是唯一的(除NULL外, NULL可能会出现多次)。
4 ALTER TABLE tbl_name ADD INDEX indexname(column_list)添加普通索引索引值可出现多次
5 ALTER TABLE tbl_name ADD FULLTEXT indexname(column_list)该语句指定了索引为 FULLTEXT,用于全文索引
```

3.2.2 删除

```
1 DROP INDEX indexname ON mytable;
```

3.2.3 查看

```
1 SHOW INDEX FROM table_name
```

3.3 索引的分类

3.3.1 普通索引

普通索引是 MySQL 中的基本索引类型, 允许在定义索引的列中插入重复值和空值。它有以下几种创建方式:

直接创建索引

```
1 CREATE INDEX index_name ON table_name(column_name)
```

这里我们以 user-profile1 为例，为 user-profile1 中的 AGE 添加索引 index
示例代码如下

```
1 CREATE INDEX index_ ON 肖宇涵.user_profile1(AGE)
```

输出结果如下

Name	Value
Updated Rows	0
Query	CREATE INDEX index_ ON 肖宇涵.user_profile1(AGE)
Finish time	Sat Dec 03 15:52:24 CST 2022

	asc Table	123 Non_unique	asc Key_name	123 Seq_in_index	asc Column_name	asc Collation	123 Cardinality	123 Sub_part	Packed
1	user_profile1	1	index_name	1	DEVICE_ID	A	1	[NULL]	[NULL]
2	user_profile1	1	index_	1	AGE	A	6	[NULL]	[NULL]

修改表结构的方式添加索引

```
1 ALTER TABLE table_name ADD INDEX index_name ON (column(length))
```

创建表的时候同时创建索引

```
1 CREATE TABLE 'table'
2 (
3 'id' int(11) NOT NULL AUTO_INCREMENT,
4 'title' char(255) CHARACTER NOT NULL,
5 'content' text CHARACTER NULL,
6 'time' int(10) NULL DEFAULT NULL,
7 PRIMARY KEY ('id'),
8 INDEX index_name (title(length))
9 )
```

3.3.2 唯一索引

唯一索引要求索引列的值必须唯一，但允许有空值。如果是组合索引，则列值的组合必须唯一。主键索引是一种特殊的唯一索引，不允许有空值。

它有以下几种创建方式：

创建唯一索引

```
1 CREATE UNIQUE INDEX indexName ON table(column(length))
```

修改表结构

```
1 ALTER TABLE table_name ADD UNIQUE indexName ON (column(length))
```

创建表的时候直接指定

```
1 CREATE TABLE 'table' (
2     'id' int(11) NOT NULL AUTO_INCREMENT,
3     'title' char(255) CHARACTER NOT NULL,
4     'content' text CHARACTER NULL,
5     'time' int(10) NULL DEFAULT NULL,
6     UNIQUE indexName (title(length))
7 );
```

3.3.3 主键索引

主键索引：是一种特殊的唯一索引，不允许有空值。（主键约束，就是一个主键索引）。一般是在建表的时候同时创建主键索引：

```
1 #随表一起建索引：
2 CREATE TABLE 'table' (
3     'id' int(11) NOT NULL AUTO_INCREMENT,
4     'title' char(255) NOT NULL,
5     PRIMARY KEY ('id')
6 );
7
8 #单独建主键索引：
9 ALTER TABLE customer add PRIMARY KEY customer(customer_no);
10 #删除建主键索引：
11 ALTER TABLE customer drop PRIMARY KEY ;
```

补充：主键索引与唯一索引的区别：

1. 主键是一种约束，唯一索引是一种索引，两者在本质上是不同的。
2. 主键创建后一定包含一个唯一性索引，唯一性索引并不一定就是主键。
3. 唯一性索引列允许空值，而主键列不允许为空值。
4. 主键索引在创建时，已经默认为非空值+唯一索引了。
5. 一个表最多只能创建一个主键索引，但可以创建多个唯一索引。
6. 主键更适合那些不容易更改的唯一标识，如自动递增列、身份证号等。
7. 主键可以被其他表引用为外键，而唯一索引不能。

3.3.4 单列索引

单列索引即一个索引只包含单个列，一个表可以有多个单列索引。

```

1 #随表一起建索引：
2 CREATE TABLE customer (id INT(10) UNSIGNED AUTO_INCREMENT ,customer_no
   VARCHAR(200),customer_name VARCHAR(200),
3 PRIMARY KEY(id),
4 KEY (customer_name)
5 );
6 #随表一起建立的索引 索引名同 列名(customer_name)
7 #单独建单值索引：
8 CREATE INDEX idx_customer_name ON customer(customer_name);
9 #删除索引：
10 DROP INDEX idx_customer_name ;

```

3.3.5 组合索引

组合索引是指在表的多个字段组合上创建的索引，只有在查询条件中使用了这些字段的左边字段时，索引才会被使用。使用组合索引时遵循最左前缀集合。

```

1 ALTER TABLE 'table' ADD INDEX name_city_age (name,city,age);

```

3.3.6 全文索引

全文索引类型为 FULLTEXT，在定义索引的列上支持值的全文查找，允许在这些索引列中插入重复值和空值。全文索引可以在 CHAR、VARCHAR 或者 TEXT 类型的列上创建。MySQL 中只有 MyISAM 存储引擎支持全文索引。InnoDB 在 mysql5.6 之后支持了全文索引

(1) 创建表的适合添加全文索引

```

1 CREATE TABLE 'table' (
2     'id' int (11) NOT NULL AUTO_INCREMENT ,
3     'title' char(255) CHARACTER NOT NULL ,
4     'content' text CHARACTER NULL ,
5     'time' int (10) NULL DEFAULT NULL ,
6     PRIMARY KEY ('id'),
7     FULLTEXT (content)
8 );

```

(2) 修改表结构添加全文索引

```
1 ALTER TABLE article ADD FULLTEXT index_content(content)
```

(3) 直接创建索引

```
1 CREATE FULLTEXT INDEX index_content ON article(content)
```

4 MySQL 中的事务

MySQL 中的事务是一组逻辑操作单元，这些操作要么全部完成，要么完全不执行。事务用于保证数据库在执行这些操作时的完整性，也就是说，如果在事务中执行的操作导致数据库状态不一致，则可以回滚事务，以便恢复数据库到事务开始前的状态。

在 MySQL 中，可以使用 **BEGIN**、**COMMIT** 和 **ROLLBACK** 语句来管理事务。

例如，假设我们有一个转账的场景，从 A 账户转账到 B 账户，并且要求 A 账户的余额减少，B 账户的余额增加。我们可以使用如下的 SQL 语句来实现：

```
1 BEGIN;
2
3 UPDATE accounts SET balance = balance - 100 WHERE name = 'A';
4 UPDATE accounts SET balance = balance + 100 WHERE name = 'B';
5
6 COMMIT;
```

在这个例子中，**BEGIN** 语句用于开始事务，**COMMIT** 语句用于提交事务，这意味着 A 账户的余额减少 100，B 账户的余额增加 100。如果在执行这两条 **UPDATE** 语句之间发生错误，则可以使用 **ROLLBACK** 语句回滚事务，以便恢复数据库到事务开始之前的状态。

还有一些其他的事务控制语句，如 **SAVEPOINT** 和 **RELEASE SAVEPOINT**，可以用于在事务中创建和删除保存点。保存点可以用于在事务中创建分支，并且可以在事务中回滚到某个特定的保存点。例如：

```
1 BEGIN;
2
3 SAVEPOINT savepoint_1;
4
5 UPDATE accounts SET balance = balance - 100 WHERE name = 'A';
6
7 SAVEPOINT savepoint_2;
8
9 UPDATE accounts SET balance = balance + 100 WHERE name = 'B';
```

```
10  
11 RELEASE SAVEPOINT savepoint_2;  
12  
13 UPDATE accounts SET balance = balance + 50 WHERE name = 'C';  
14  
15 ROLLBACK TO SAVEPOINT savepoint_1;  
16  
17 COMMIT;
```

在这个例子中，我们使用 SAVEPOINT 语句创建了两个保存点 savepoint-1 和 savepoint-2，然后执行了几条 UPDATE 语句。RELEASE SAVEPOINT 语句用于删除 savepoint-2 保存点，这意味着我们无法再回滚到 savepoint-2。最后，我们使用 ROLLBACK TO SAVEPOINT 语句回滚到 savepoint-1，这意味着 A 账户的余额没有发生变化，B 账户的余额也没有发生变化，但是 C 账户的余额增加了 50。