实验十五 索引与数据库备份

姓名	学号	学院	日期
臧祝利	202011998088	人工智能学院	2022.11.28

实验目的

(1) 了解什么是索引；

(2) 掌握创建索引的方法和技巧，熟悉如何删除索引；

(3) 了解数据备份和还原的概念；

(4) 掌握各种数据备份和还原的方法；

(5) 掌握加密算法；

实验内容

(1) 创建索引：使用 SQL 语句创建唯一索引、普通索引、组合索引；重命名索引；删除索引；

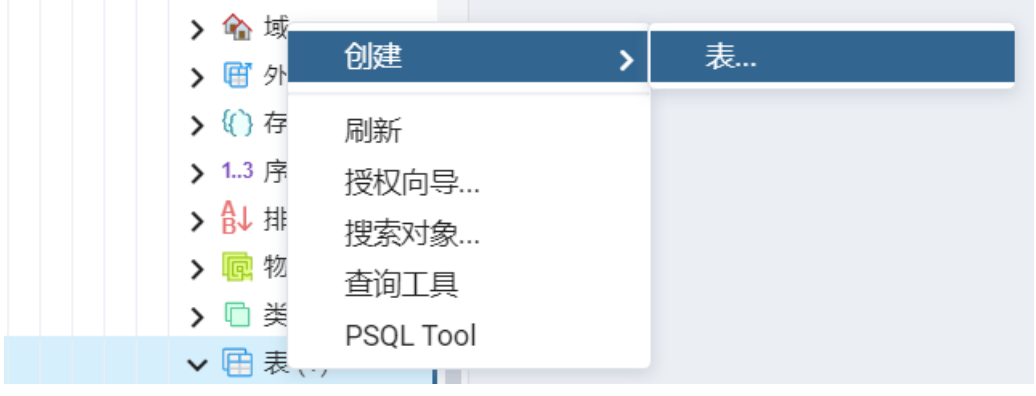
(2) 数据备份与数据还原；

(3) 加密算法

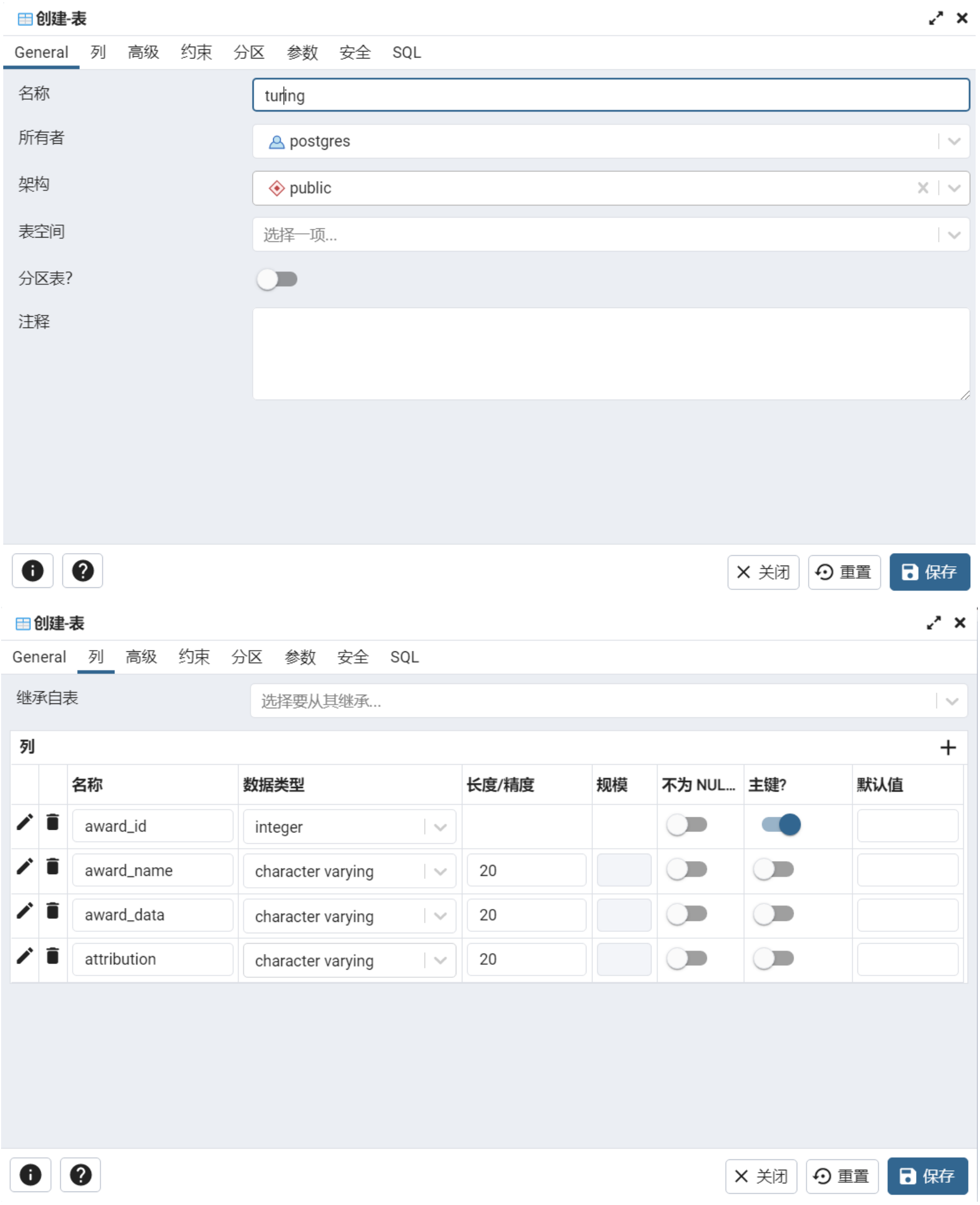
实验思路

(1) 在数据库turingaward 中创建数据表 turing。

Step1. 创建数据库 `Labs15`，选择其“架构”，选择其中的“表”，右击选择新建表；



Step2. 设置表的属性；



(2) 向turing表中插入数据。

Step1. 右击`turing`表，选择查看/编辑数据中的所有行；

Step2. 添加数据，点击①添加行，点击②进行保存；

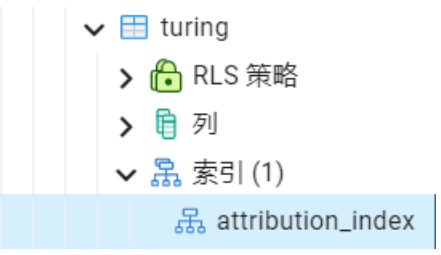
	award_id [PK] integer	award_name character varying (20)	award_data character varying (20)	attribution character varying (20)
1+	1	Charles W. Bachman	1973	network database
2+	2	Edgar F. Codd	1981	Relational database
3+	3	James Gray	1998	Transaction
4+	4	Michael Stonebraker	2014	Modern Database
5+	5	Yann LeCun	2018	deep learning
6+	6	Tim Berners-Lee	2016	web

(3) 在贡献领域attribution上创建普通索引attribution_index。

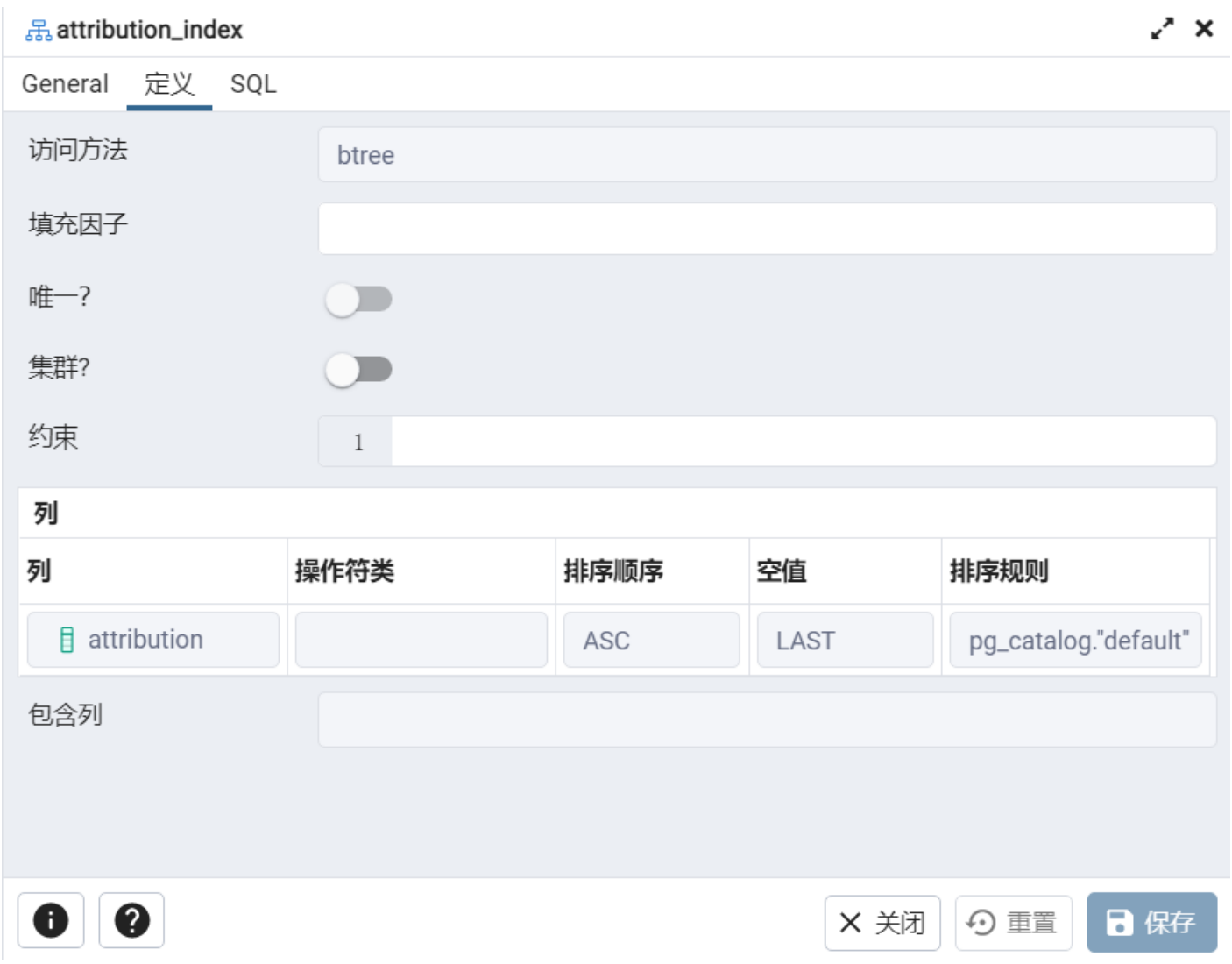
输入以下代码：

```
create index attribution_index on turing(attribution);
```

若成功，如图所示：



右击，选择属性，查看定义信息；

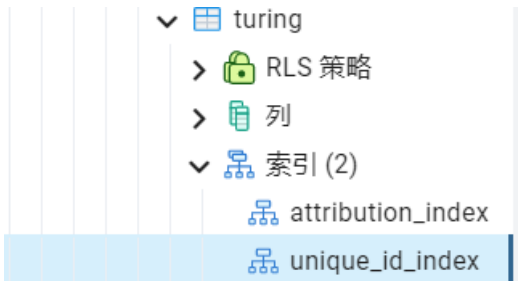


(4) 在award_id上创建唯一索引unique_id_index。

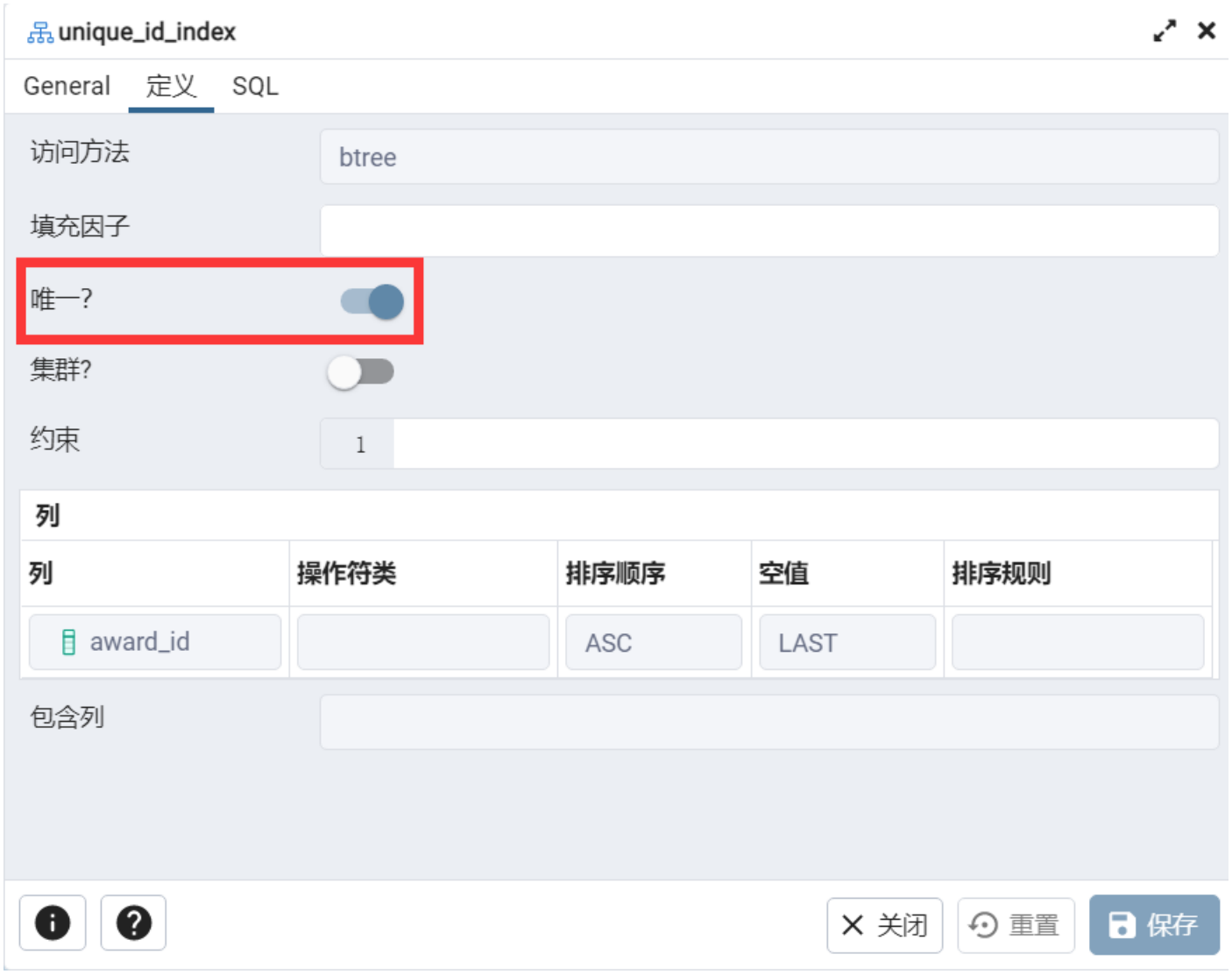
执行代码：

```
create unique index unique_id_index on turing(award_id);
```

成功后如图所示：



右击查看属性，唯一性被标注；

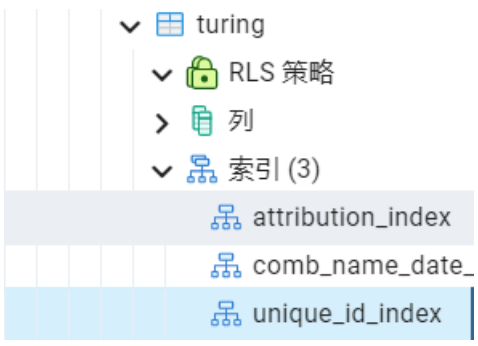


(5) 在award_name和award_date上创建组合索引comb_name_date_index。

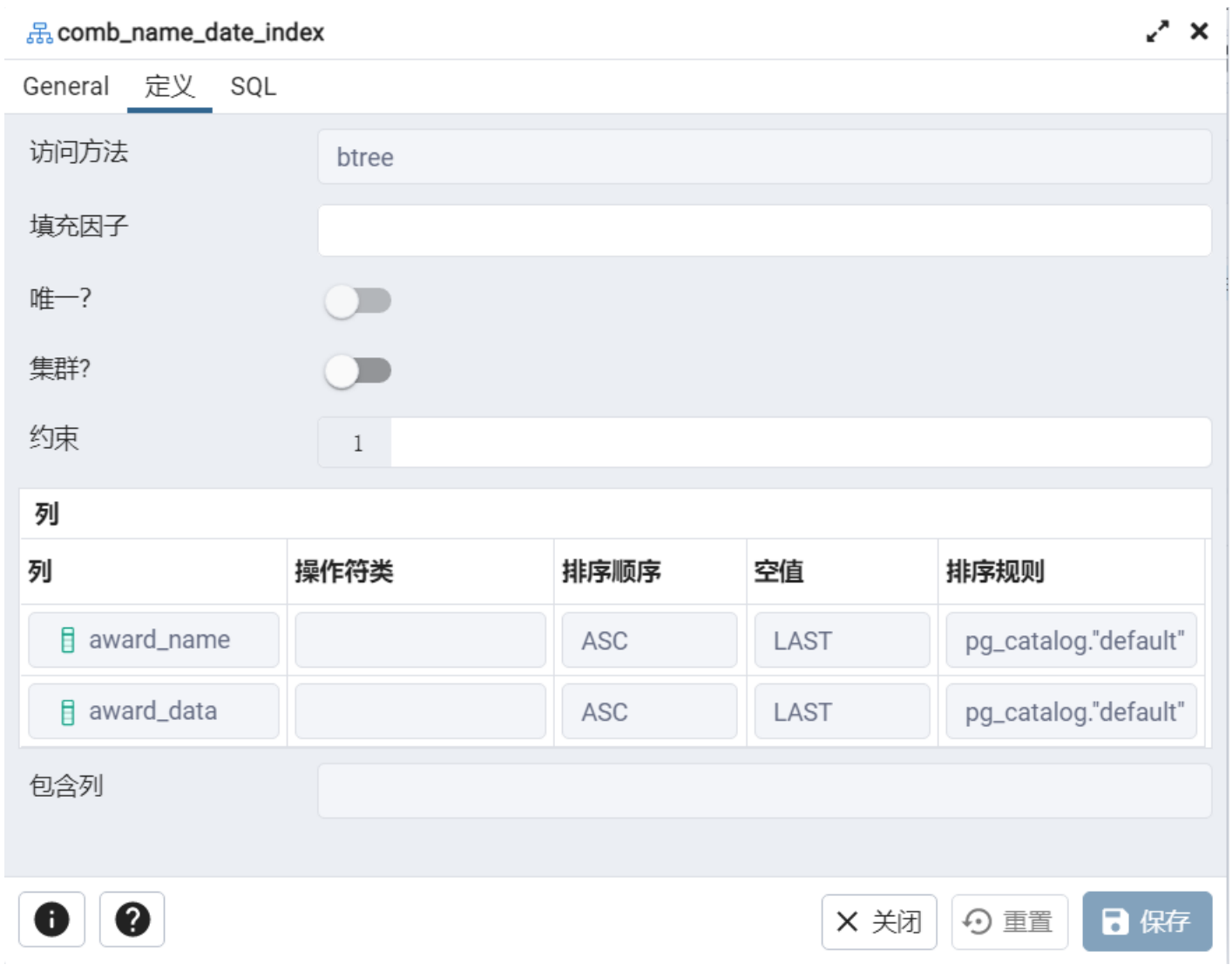
执行代码：

```
create index comb_name_date_index on turing(award_name,award_data);
```

执行成功如图所示：



右击查看属性：

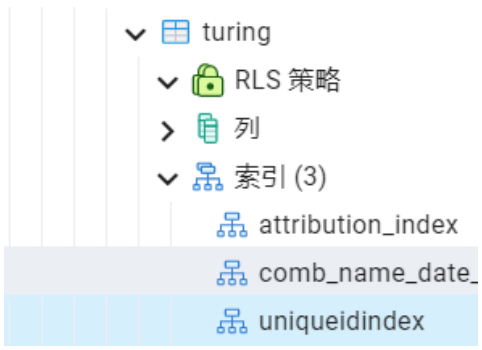


(6) 将索引unique_id_index重命名为uniqueidindex。

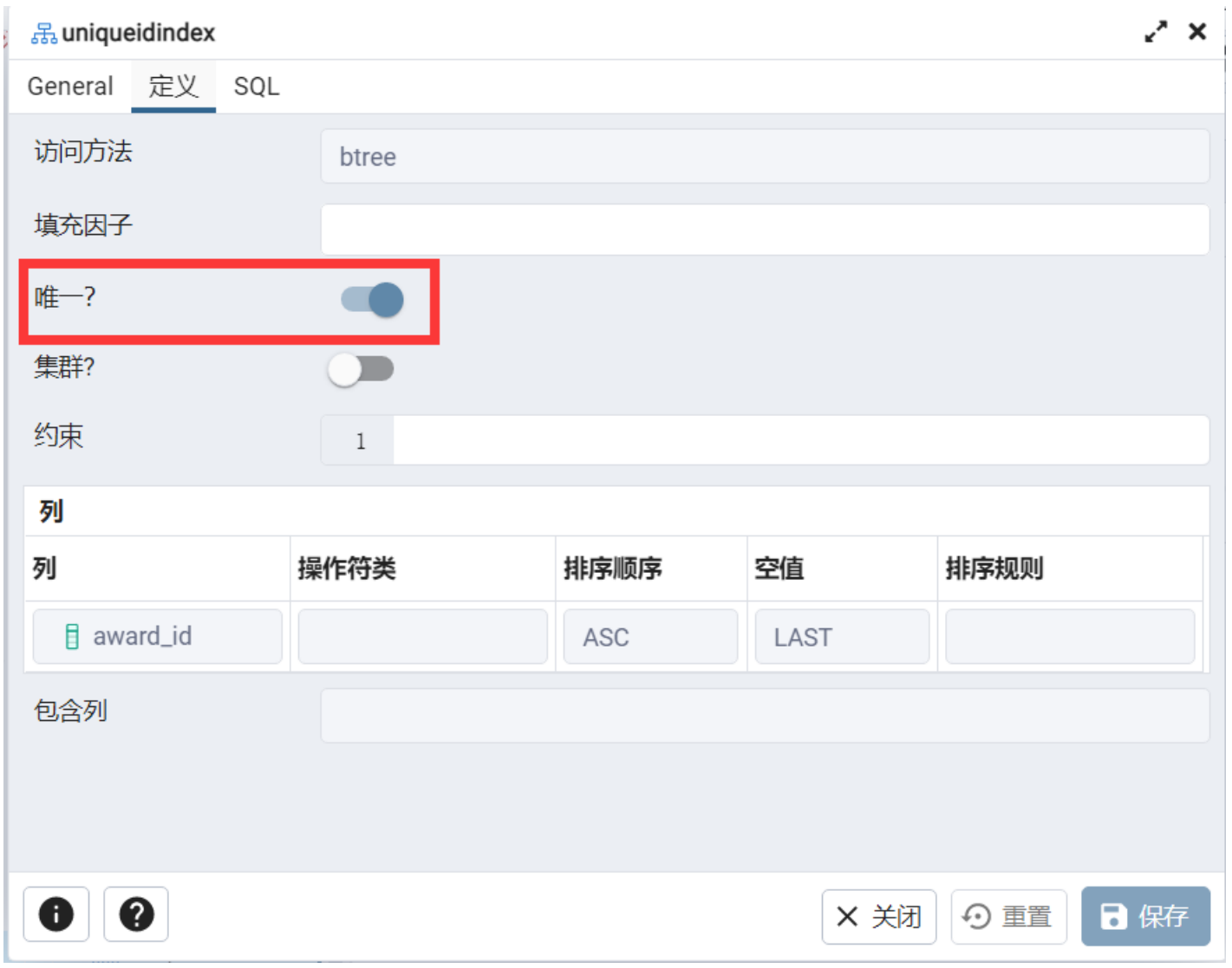
执行代码：

```
alter index unique_id_index rename to uniqueidindex;
```

执行成功后：



右击查看属性：

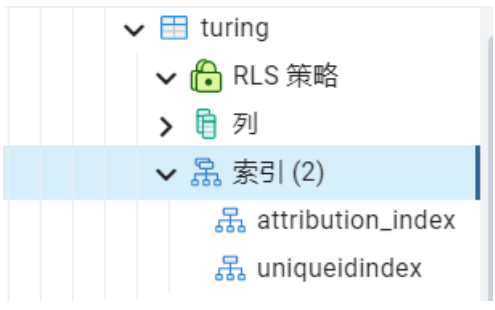


(7) 将索引comb_name_date_index删除。

执行代码：

```
drop index comb_name_date_index;
```

运行结果如下：



(8) 查询turing表中attribution为“web”的信息，并显示执行命令和显示实际运行时间。

执行代码:

```
explain analyse SELECT * FROM turing
WHERE attribution = 'web';
```

结果如下：

	QUERY PLAN text
1	Seq Scan on turing (cost=0.00..1.07 rows=1 width=178) (actual time=0.034..0.037 rows=1 loops=1)
2	Filter: ((attribution)::text = 'web'::text)
3	Rows Removed by Filter: 5
4	Planning Time: 4.403 ms
5	Execution Time: 0.068 ms

(9) 使用索引查询turing表中attribution为“web”的信息，并与问题(8)的查询速度做比较。

执行代码：

```
explain SELECT * FROM turing
WHERE attribution = 'web';
```

执行结果：

	QUERY PLAN text
1	Seq Scan on turing (cost=0.00..1.07 rows=1 width=178)
2	Filter: ((attribution)::text = 'web'::text)

并没有使用索引，因为表太小，规划器认为不需要索引扫描，因此强制使用索引进行查询；

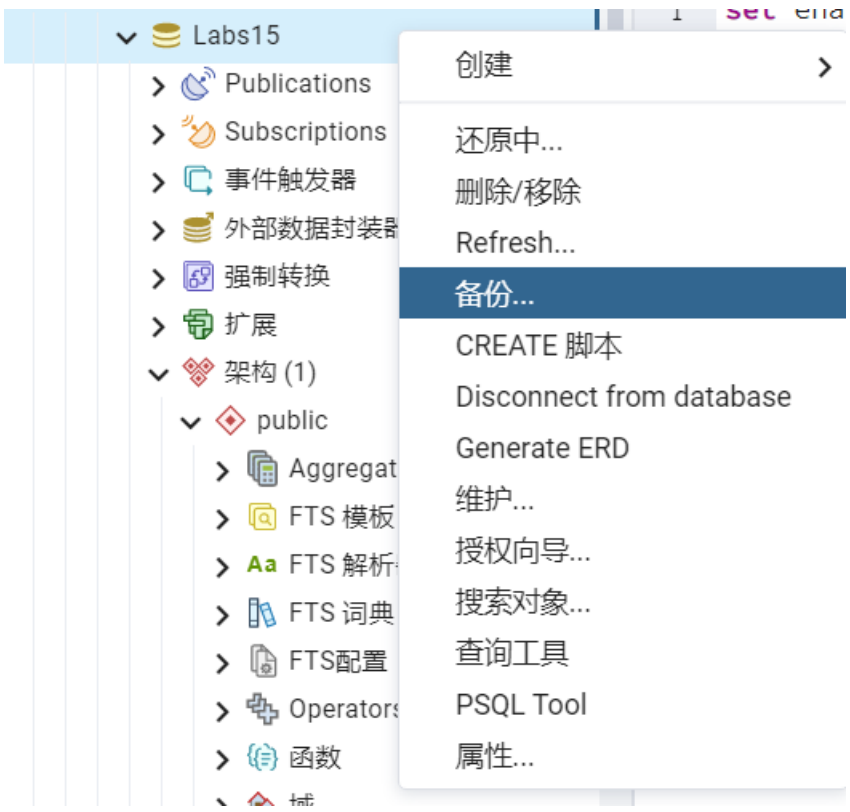
执行代码：

	QUERY PLAN text
1	Index Scan using attribution_index on turing (cost=0.13..8.15 rows=1 width=178)
2	Index Cond: ((attribution)::text = 'web'::text)

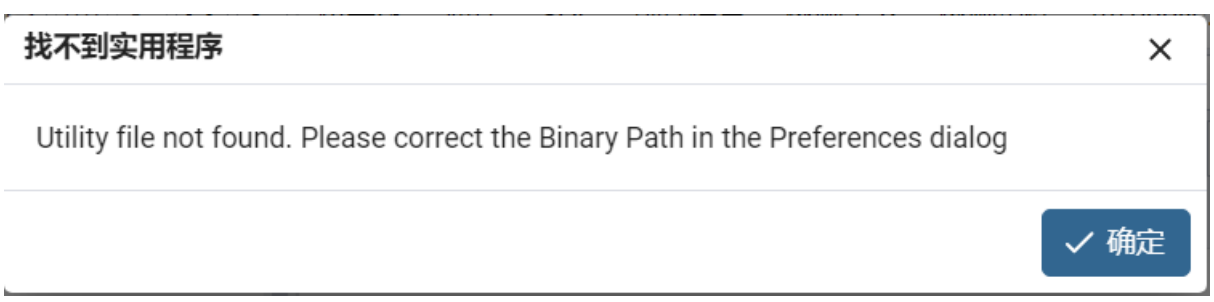
可以发现其中`cost`增加，说明使用索引扫描时间增加，原因是表太小，增加的cpu时间显得更多

(10) 用 pgAdmin 备份数据库 `exam_system`，删除数据库，并使用 pgAdmin 将其还原。

Step1. 右击数据库，点击“备份”；

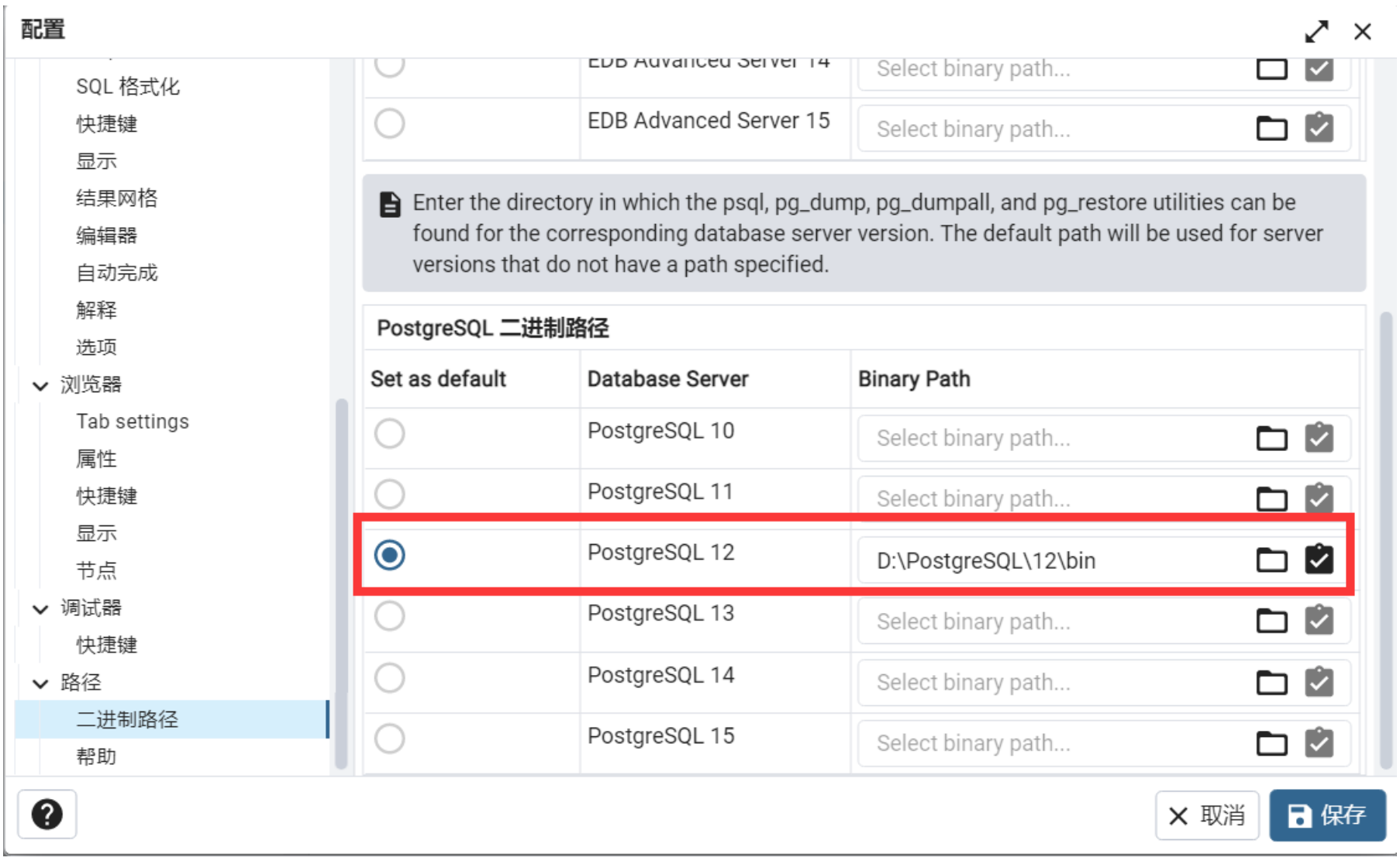


出现错误！



查询后，发现需要更改路径，选择文件

选择“路径”，“二进制路径”，将postgresql的路径选中，保存；

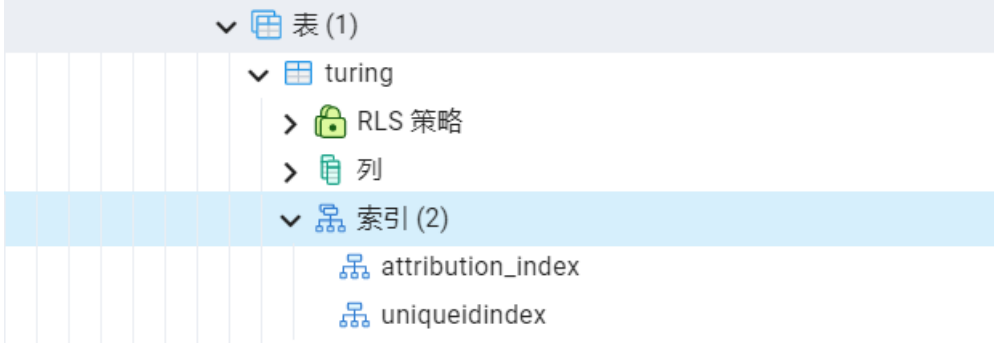


再次备份，成功，输入文件名；

Step2. 恢复数据库，新建一个空数据库 `Labs15_copy`；

选择文件路径：

查看数据库内容，成功恢复；



(11) 对字符串“I love database”使用MD5加密算法，返回加密结果；

先执行如下代码：

```
create extension pgcrypto;
```

再使用MD5进行加密：

```
SELECT MD5('I love database');
```

结果如下：

	md5 text
1	3de545785f0378bb3d5129df19c96fbf

(12) 用AES算法、以字符串“datebase”做密钥，对“abcdef”的对称加密，输出加密密文，再对密文进行解密，还原出“abcdef”。

执行以下代码进行加密：

```
SELECT encode(encrypt('abcdef','datebase','aes'),'hex');
```

加密结果如下：

	encode text
1	83b78dcbd0503d3420de40ae0ee6ee6d

执行以下代码进行解密：

```
SELECT convert_from(decrypt(decode('83b78dcbd0503d3420de40ae0ee6ee6d','hex'),'datebase','aes'),'SQL_ASCII');
```

解密结果如下：

	convert_from text
1	abcdef

实验结果

见实验思路

源代码

见实验思路

思考题

（1）索引对数据库性能如此重要，我们应该如何使用它。

索引具有以下优点：

- 大大加快数据的检索速度
- 加速表和表之间的连接
- 使用分组和排序子句进行数据检索时，可以显著减少查询中分组和排序的时间；
- 通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性；

适合创建索引的字段：

- 经常作查询选择的字段
- 经常作表连接的字段
- 经常出现在`order by`，`group by`，`distinct`后面的字段

索引的缺点：

- 创建索引和维护索引需要耗费时间，对表中的数据进行增加、删除和修改时索引也需要动态维护，降低了数据的维护速度；
- 索引需要占物理空间；

需要注意的是，**索引不是越多越好，不要对经常变动的数据加索引，小数据的表不需要索引**；

以下情况不需要添加索引：

- 在查询中很少使用或者很少参考的列不应该创建索引；
- 只有很少数据值的列不应该增加索引；
- 当修改性能远远大于检索性能时，不应该创建索引；

（2）使用 pdAdmin 恢复数据库时需要注意哪些问题？

就我的实验过程而言，首先恢复数据库时要先进行备份，备份时会出现 错误(ctrl+单击可跳转)：`Utility file not found. Please correct the Binary Path in the Preferences dialog`错误，原因是在 pgAdmin4 中，没有指向 PostgreSQL DBMS 的可执行文件 psql 的路径，导致无法执行操作 postgresql 的命令或语句。

因此需要将postgresql的bin文件所在路径放入设置中，即可以解决问题；

恢复时，关键是找到文件所在的位置；整体感觉除了备份时遇到的错误，没有什么特别需要注意的地方~😄