1. 模仿PPT中的建表插值过程（id INT, val INT）（数据量大约10,000即可）。

```
create table index_test (
    id int,
    val int
);

delimiter //

create procedure insert_index_test ()
begin
  declare i int default 16210000;
  while i < 16220000 do
    insert into index_test values (i, rand() * 100);
    set i = i + 1;
  end while;
end;//

delimiter ;

call insert_index_test();

select count(*) from index_test;
```

```
+----------+
| count(*) |
+----------+
|    10000 |
+----------+
1 row in set (0.01 sec)
```

```
explain
    select val
    from index_test
    where id < 16210050
    order by id\G
```

```
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: index_test
   partitions: NULL
         type: ALL
```

```
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 10157
     filtered: 33.33
        Extra: Using where; Using filesort
1 row in set, 1 warning (0.00 sec)
```

```
create unique index id_uni_idx
    on index_test (id);

explain
    select val
    from index_test
    where id < 16210050
    order by id\G
```

```
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: index_test
   partitions: NULL
         type: range
possible_keys: id_uni_idx
          key: id_uni_idx
      key_len: 5
          ref: NULL
         rows: 50
     filtered: 100.00
        Extra: Using index condition
1 row in set, 1 warning (0.00 sec)
```

2. 建立id升序同时val降序的非聚类索引。

```
create index id_val_idx
    on index_test (id, val desc);
```

3. 分别至少写两个命中和不命中这个索引的查询。

命中：

```
-- where中使用了val
select *
from index_test
where val = 1;
```

```
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: index_test
   partitions: NULL
         type: index
possible_keys: id_val_idx
          key: id_val_idx
      key_len: 10
          ref: NULL
         rows: 10157
     filtered: 10.00
        Extra: Using where; Using index
1 row in set, 1 warning (0.00 sec)
```

```
-- where中使用了id和val
select *
from index_test
where id = 1 or val = 1;
```

```
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: index_test
   partitions: NULL
         type: index
possible_keys: id_uni_idx,id_val_idx
          key: id_val_idx
      key_len: 10
          ref: NULL
         rows: 10157
     filtered: 10.01
        Extra: Using where; Using index
1 row in set, 1 warning (0.00 sec)
```

不命中：

```
-- select中只有id，没有where条件
select id
from index_test;
```

```
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: index_test
   partitions: NULL
         type: index
```

```
possible_keys: NULL
          key: id_uni_idx
      key_len: 5
          ref: NULL
         rows: 10157
     filtered: 100.00
        Extra: Using index
1 row in set, 1 warning (0.00 sec)
```

```
-- where中只用了id，并且在可能的范围内，会命中第一个索引
select *
from index_test
where id = 16210050;
```

```
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: index_test
   partitions: NULL
         type: const
possible_keys: id_uni_idx,id_val_idx
          key: id_uni_idx
      key_len: 5
          ref: const
         rows: 1
     filtered: 100.00
        Extra: NULL
1 row in set, 1 warning (0.01 sec)
```

4. 写注释总结一下大概什么样的查询能命中这个索引。

- where 子句中包含 id 或 val
- select 子句中包含 id 或 val
- 不存在其他更合适的索引，比如 id 上的唯一索引

5. 建立触发器，插入数据后，若val值为完全平方数（1,4,9...）时打印“Good”，自拟测试数据检测触发器是否生效。

```
delimiter //

create
  trigger insert_square
  after insert
  on index_test for each row
  begin
    declare temp int default sqrt(new.val);
    if temp * temp = new.val then select 'Good' into @result;
    end if;
  end;//

delimiter ;
```

```
mysql> select @result;
+------------------+
| @result          |
+------------------+
| NULL             |
+------------------+
1 row in set (0.00 sec)
```

```
insert into index_test values (10000, 4);
```

```
mysql> select @result;
+---------+
| @result |
+---------+
| Good    |
+---------+
1 row in set (0.00 sec)
```

```
set @result = NULL;

insert into index_test values (10001, 5);
```

```
mysql> select @result;
+---------+
| @result |
+---------+
| NULL    |
+---------+
1 row in set (0.00 sec)
```