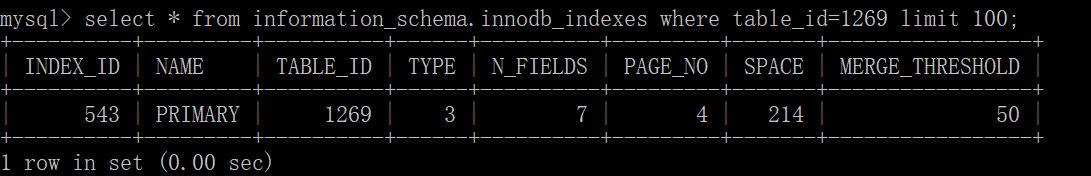
Atitit impt db insert data process 插入数据流程

* 插入数据 向表中插入一条数据如下

insert into world.user(name,age,gender,create\_time) values('木叶潇潇',18,1,now())

从sql中提取数据库名和表名，从information\_schema.innodb\_tables中查出表id

根据表id，从information\_schema.innodb\_indexes中查出表对应的聚集索引的Root Page No 为4。 

通过Root Page No 4计算出Root Page的物理地址。根据Root Page中指定的段信息，向Root Page中插入索引数据，向数据段对应的页中插入数据行，并关联两种类型的页。

如果一页空间不足，会计算出当前页所在的区/簇并向其申请空间，区/簇则会根据 XDES Entry中的bitmap查询空闲的页并进行分配。如果区/簇也没有空闲空间，则会一级一级向上面的段、表空间、操作系统申请所需空间。

申请到的表空间会存储在各自对应的链表中（如：表空间申请到的空间会存储在对应的FSP\_FREE链表中）。

在页分配或扩展时，为了保证通过innodb\_indexes中的Root Page No能找到它，Root Page物理空间与B+树对应的Root 节点保持不变,即页号不变，永远是页号为4的那块空间。

当B+对应的物理页不断变化时，为了保证树的平衡，会产生新的Root节点，为了保持Root页不变，innodb是通过交换的方式，把新的Root节点数据复制交换到原来的Root Page页，这样就可以保证Root Page永远不变，即保证表与物理空间的关联永远不会断开。

## 事务流程很复杂，要写事务开启，undo，op，redo，日志，事务提交，多增加类5部操作

为了记录行本身的状态，一条记录innodb会增加额外的记录头信息。如果是叶子节点，还会增加：row\_id（隐藏的主键）、trx\_id（事务id）、回滚指针等附加字段。

## 索引增加