期刊表： paper

```
CREATE TABLE public.paper
(
  id serial NOT NULL,
  title character varying(1024) NOT NULL,
  sizekb integer NOT NULL default 0,
  fulltext character varying(1024),
  abstr character varying(204800),
  keyword character varying(102400),
  abstr_cn character varying(204800),
  keyword_cn character varying(102400),
  doi character varying(1024),
  nihms_id integer,

  source character varying(255),
  url_abstract character varying(10248),
  url_fulltext character varying(10248),
  source_id integer,
  source_id_str character varying(255),
  pm_id integer,

  journal character varying(1024) NOT NULL,
  journal_year integer NOT NULL,
  journal_volume character varying(1024),
  journal_no character varying(255),
  page_begin integer NOT NULL default 0,
  page_end integer NOT NULL default 0,
  online_date timestamp without time zone,

  authors character varying(102400),
  author_orgs character varying(204800),

  ts timestamp without time zone NOT NULL,
  CONSTRAINT paper_pkey PRIMARY KEY (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.paper
  OWNER TO postgres;
```

sizekb：全文文件的大小，单位kb。没有为0。默认也为0.
fulltext：全文路径。
doi： 有些paper没有doi。
nihms_id：似乎是美国医学期刊编号。大部分情况下没有。
source: 来源，有pmc、cnki、wanfang，magtech.com.cn等。这里统一写来源的 主域名，不带二级域名的。
url： 对应的爬取链接的完整url，方便校对。
source_id：来源处的id
source_id_str：来源处的id，string类型
pm_id：pubmed id。
journal：期刊名字
journal_year: 期刊年
journal_volume: 期刊卷，一般卷和年一一对应。
journal_no：期刊的期。 年内第几次出版

期刊唯一标示表： medid

```
CREATE TABLE public.medid
(
    mid character varying(1224) NOT NULL,
    midhash integer NOT NULL,
    paper_id integer NOT NULL,
    CONSTRAINT medid_pkey PRIMARY KEY (mid)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.medid
  OWNER TO postgres;
```

mid: 每篇paper的唯一id，需要据此去重。计算规则是 journal-year-journal_no-title -> 去掉所有空格。
midhash: mid计算得到的hash值。
paper_id: 对应paper表中的id。
常见用法：
计算文章是否重复，通过比较 midhash==$1 && mid=$2 得到 paper_id。 注意一定要先比较midhash，利用短路的性能。

期刊附件、引文表： paper_asset

```
CREATE TABLE public.paper_asset
(
  id serial NOT NULL,
  img_path character varying(255) NOT NULL,
  pmc_id integer NOT NULL,
  status integer NOT NULL,
  ts timestamp without time zone NOT NULL,
  CONSTRAINT paper_asset_pkey PRIMARY KEY (id),
  CONSTRAINT uk_ssqx2upil90n946aokwby7gqw UNIQUE (pmc_id, img_path)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.paper_asset
  OWNER TO postgres;
```

爬取日志表：paper_log

```
CREATE TABLE public.paper_log
(
id serial NOT NULL,
paper_id integer,
source character varying(255),
url character varying(10248),
status character(1) NOT NULL default 'y',
complete integer NOT NULL,
status_msg character varying(255),
ts timestamp without time zone,
CONSTRAINT paper_log_pkey PRIMARY KEY (id),
CONSTRAINT uk_5qvw05v9bd1jw1fl381s3u4pk UNIQUE (url)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.paper_log
  OWNER TO postgres;
```

paper_id：paper中的id，可能没有！因为如果是失败了，可能就不会插入到paper中，所以这个时候就没有对应的pid了。
source: 来源，有pmc、cnki、wanfang，magtech.com.cn等。这里统一写来源的 主域名，不带二级域名的。
url： 对应的爬取链接的完整url，方便校对。
status：y: OK　n: No content found　e: error found
complete：0 : 未进行爬取过； 1： 已获得摘要； 3：已获取全文； 7： 已push到ES引擎中
status_msg：描述。一般用于出错时候的填些补充信息。
ts: 时间戳

常用操作：
根据status=n 或 e的，进行二次爬取。
根据complete = 1，去获取全文； 根据complete=3，去push到ES引擎中。

期刊作者表：paper_author

```
CREATE TABLE public.paper_author
(
  id serial NOT NULL,
  paper_id integer NOT NULL,
  author_name character varying(255) NOT NULL,
  author_email character varying(255),
  status character(1) NOT NULL default 'y',
  extra character varying(2048),
  CONSTRAINT paper_author_pkey PRIMARY KEY (id),
  CONSTRAINT uk_ssqx2upil902346aokwby7gqw UNIQUE (paper_id, author_name)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.paper_author
  OWNER TO postgres;
```

status : y/n/e 参见paper_log
author_email： 可为空。但author_name非空。需要注意，当email不为空时候，一定得有个author_name，如果没有，确定使用最后一个作者。
extra： 当status不等于y时，把相关文本存放在这个字段里。