

# Insolvenzkultur in Preußen und im Deutschen Kaiserreich (1855–1914)

Elektronischer Anhang

Inauguraldissertation  
zur Erlangung des akademischen Grades eines  
Doktors der Rechte der Universität Mannheim  
eingereicht von Christoph Kling

2019

## Inhaltsübersicht

F. Anhang.....	1
II. Konkurshistorische Datenbank .....	1
11. Datenstruktur (SQL).....	1
a. Datenbank.....	1
b. Tabellen.....	1
c. Ansichten .....	3
d. Indizes .....	8
e. Fremdschlüsselbeschränkungen .....	9
12. Liste der Datenbank-Abfragen (SQL).....	10
13. Liste der Statistiksoftware-Anweisungen (Stata).....	19

## F. Anhang

### II. Konkurshistorische Datenbank

#### 11. Datenstruktur (SQL)

##### a. Datenbank

```
-- Datenbank: reichsanzeiger
CREATE
DATABASE reichsanzeiger DEFAULT CHARACTER SET
latin1 COLLATE latin1_swedish_ci;
```

##### b. Tabellen

```
-- Tabellenstruktur für Tabelle k_eroef
CREATE
TABLE
k_eroef ( id INT(11) NOT NULL COMMENT 'Konkurseröffnungsidentifikationsnummer
(KID)',
aid INT(11) NOT NULL COMMENT 'Konkursbild-ID',
pid INT(11) DEFAULT NULL COMMENT 'übergeordnete KID',
s_name VARCHAR(1024) NOT NULL COMMENT 'Schuldnername',
s_beruf VARCHAR(255) NOT NULL COMMENT 'Schuldnerberuf',
s_ort VARCHAR(255) NOT NULL COMMENT 'Schuldnerort',
v_name VARCHAR(1024) NOT NULL COMMENT 'Verwaltername',
v_beruf VARCHAR(255) NOT NULL COMMENT 'Verwalterberuf',
v_ort VARCHAR(255) NOT NULL COMMENT 'Verwalterort',
vid INT(11) DEFAULT NULL COMMENT 'Verwalteridentifikationsnummer (VID)',
eroef_dat DATE NOT NULL COMMENT 'Eröffnungsdatum',
bek_dat DATE NOT NULL COMMENT 'Bekanntmachungsdatum',
anz_dat DATE NOT NULL COMMENT 'Anzeigefrist',
anm_dat DATE NOT NULL COMMENT 'Anmeldefrist',
gvers_dat DATE NOT NULL COMMENT 'Gläubigerversammlung',
pruef_dat DATE NOT NULL COMMENT 'Prüfungstermin',
gericht VARCHAR(255) NOT NULL COMMENT 'Konkursgericht',
t_dat DATE NOT NULL COMMENT 'Todesdatum',
flucht_k tinyint(1) NOT NULL COMMENT 'auf Flucht',
haft_k tinyint(1) NOT NULL COMMENT 'in Haft',
abwesend_k tinyint(1) NOT NULL COMMENT 'abwesend',
unbaufent_k tinyint(1) NOT NULL COMMENT 'unbekannter Aufenthalt',
nachlass_k tinyint(1) NOT NULL COMMENT 'Nachlass',
ohg_k tinyint(1) NOT NULL COMMENT 'OHG',
kg_k tinyint(1) NOT NULL COMMENT 'KG',
gesellter_k tinyint(1) NOT NULL COMMENT 'Gesellschafter',
ag_k tinyint(1) NOT NULL COMMENT 'AG',
kga_a_k tinyint(1) NOT NULL COMMENT 'KGaA',
eg_k tinyint(1) NOT NULL COMMENT 'eG',
verein_k tinyint(1) NOT NULL COMMENT 'Verein',
vvag_k tinyint(1) NOT NULL COMMENT 'VVaG',
gmbh_k tinyint(1) NOT NULL COMMENT 'GmbH',
sgs_k tinyint(1) NOT NULL COMMENT 'sonstige Gesellschaft',
firma_k tinyint(1) NOT NULL COMMENT 'Firma',
liq_k tinyint(1) NOT NULL COMMENT 'in Liquidation',
mj_k tinyint(1) NOT NULL COMMENT 'minderjährig',
gu_k tinyint(1) NOT NULL COMMENT 'geschäftsunfähig',
w_k tinyint(1) NOT NULL COMMENT 'weiblich',
getr_k tinyint(1) NOT NULL COMMENT 'getrennt',
eheteute_k tinyint(1) NOT NULL COMMENT 'Eheteute',
gg_k tinyint(1) NOT NULL COMMENT 'Gütergemeinschaft',
gt_k tinyint(1) NOT NULL COMMENT 'Gütertrennung',
bemerck text NOT NULL COMMENT 'Bemerkung',
LAST VARCHAR(255) NOT NULL,
ts TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON
```

```

UPDATE
CURRENT_TIMESTAMP ) ENGINE = InnoDB DEFAULT CHARSET = latin1;
-- Tabellenstruktur für Tabelle k_beend
CREATE
TABLE
k_beend ( id INT(11) NOT NULL,
aid INT(11) NOT NULL,
pid INT(11) DEFAULT NULL,
kid INT(11) DEFAULT NULL,
s_name VARCHAR(1024) DEFAULT NULL,
s_beruf VARCHAR(255) DEFAULT NULL,
s_ort VARCHAR(255) DEFAULT NULL,
gericht VARCHAR(255) DEFAULT NULL,
typ enum('s',
'z',
'm',
'a',
'e',
'u') NOT NULL COMMENT 's = Schlussverteilung; z = Zwangsvergleich; m =
mangels Masse; a = allgemeine Zustimmung; e = Eröffnungsbeschluss aufgehoben; u =
unbekannt',
aufh_dat DATE NOT NULL COMMENT 'Aufhebungsdatum',
bek_dat DATE NOT NULL COMMENT 'Bekanntmachungsdatum',
t_dat DATE NOT NULL COMMENT 'Todesdatum',
flucht_k tinyint(1) NOT NULL,
haft_k tinyint(1) NOT NULL,
abwesend_k tinyint(1) NOT NULL,
unbaufent_k tinyint(1) NOT NULL,
nachlass_k tinyint(1) NOT NULL,
ohg_k tinyint(1) NOT NULL,
kg_k tinyint(1) NOT NULL,
gesellter_k tinyint(1) NOT NULL,
ag_k tinyint(1) NOT NULL,
kga_k tinyint(1) NOT NULL,
eg_k tinyint(1) NOT NULL,
verein_k tinyint(1) NOT NULL,
vvag_k tinyint(1) NOT NULL,
gmbh_k tinyint(1) NOT NULL,
sgs_k tinyint(1) NOT NULL,
firma_k tinyint(1) NOT NULL,
liq_k tinyint(1) NOT NULL,
mj_k tinyint(1) NOT NULL,
gu_k tinyint(1) NOT NULL,
w_k tinyint(1) NOT NULL,
getr_k tinyint(1) NOT NULL,
ehelute_k tinyint(1) NOT NULL,
gg_k tinyint(1) NOT NULL,
gt_k tinyint(1) NOT NULL,
eaez_k tinyint(1) NOT NULL COMMENT 'Eröffnung außerhalb Erfassungszeitraum',
bemerk text NOT NULL,
LAST VARCHAR(255) NOT NULL,
ts TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE
CURRENT_TIMESTAMP ) ENGINE = InnoDB DEFAULT CHARSET = latin1;
-- Tabellenstruktur für Tabelle nberuf
CREATE
TABLE
nberuf ( id INT(11) NOT NULL,
beruf VARCHAR(255) NOT NULL,
berufn VARCHAR(255) NOT NULL,
bemerk text NOT NULL ) ENGINE = InnoDB DEFAULT CHARSET = latin1;
-- Tabellenstruktur für Tabelle ngericht
CREATE
TABLE
ngericht ( id INT(11) NOT NULL,
gericht VARCHAR(255) NOT NULL,
gerichtn VARCHAR(255) NOT NULL,
bemerk text NOT NULL ) ENGINE = InnoDB DEFAULT CHARSET = latin1;
-- Tabellenstruktur für Tabelle nort

```

```

CREATE
TABLE
  nort ( id INT(11) NOT NULL,
  ort VARCHAR(255) NOT NULL,
  ortn VARCHAR(255) NOT NULL,
  bemerk text NOT NULL ) ENGINE = InnoDB DEFAULT CHARSET = latin1;
-- Tabellenstruktur für Tabelle nregiogericht
CREATE
TABLE
  nregiogericht ( id INT(11) NOT NULL,
  gerichtn VARCHAR(255) NOT NULL,
  regiogericht VARCHAR(255) NOT NULL,
  bemerk text NOT NULL ) ENGINE = InnoDB DEFAULT CHARSET = latin1;
-- Tabellenstruktur für Tabelle tgerichtort
CREATE
TABLE
  tgerichtort ( id INT(11) NOT NULL,
  gerichtn VARCHAR(255) NOT NULL,
  ortn VARCHAR(255) NOT NULL,
  bemerk text NOT NULL ) ENGINE = InnoDB DEFAULT CHARSET = latin1;
-- Tabellenstruktur für Tabelle titelbl
CREATE
TABLE
  titelbl ( id INT(11) NOT NULL,
  fid INT(11) NOT NULL,
  bildnr INT(11) NOT NULL,
  ausgabe INT(11) DEFAULT NULL,
  jahr INT(11) DEFAULT NULL,
  ts TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON
  UPDATE
  CURRENT_TIMESTAMP ) ENGINE = InnoDB DEFAULT CHARSET = latin1;
-- Tabellenstruktur für Tabelle verw
CREATE
TABLE
  verw ( id INT(11) NOT NULL,
  berufnm VARCHAR(255) DEFAULT NULL COMMENT 'manuelle Festlegung des normierten
  Berufs',
  bemerk text NOT NULL ) ENGINE = InnoDB DEFAULT CHARSET = latin1;
-- Tabellenstruktur für Tabelle `filme`
CREATE
TABLE
  `filme` ( `id` INT(11) NOT NULL,
  `filmnr_kurz` INT(11) NOT NULL,
  `filmnr_lang` INT(11) NOT NULL,
  `ts` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON
  UPDATE
  CURRENT_TIMESTAMP ) ENGINE = InnoDB DEFAULT CHARSET = latin1;

```

### c. Ansichten

```

-- Struktur des Views k_beend_cn
CREATE
ALGORITHM = MERGE DEFINER = reichsanzeiger@localhost SQL SECURITY INVOKER VIEW
k_beend_cn AS SELECT
  b.`id` AS `id`,
  b.aid AS aid,
  b.pid AS pid,
  b.kid AS kid,
  b.s_name AS s_name,
  b.s_beruf AS s_beruf,
  b.s_ort AS s_ort,
  b.gericht AS gericht,
  COALESCE(ngg.gerichtn,
  b.gericht) AS gericht_cn,
  COALESCE(tgerichtort.ortn,
  COALESCE(ngg.gerichtn,
  b.gericht)) AS gerichtort_cn,
  COALESCE(nregiogericht.regiogericht,
  COALESCE(ngg.gerichtn,

```

```

b.gericht)) AS regiogericht_cn,
b.typ AS typ,
b.aufh_dat AS aufh_dat,
IF((b.aufh_dat = '0000-00-00'),
b.bek_dat,
b.aufh_dat) AS aufh_dat_c,
b.bek_dat AS bek_dat,
b.t_dat AS t_dat,
b.flucht_k AS flucht_k,
b.haft_k AS haft_k,
b.abwesend_k AS abwesend_k,
b.unbaufent_k AS unbaufent_k,
b.nachlass_k AS nachlass_k,
b.ohg_k AS ohg_k,
b.kg_k AS kg_k,
b.gesellter_k AS gesellter_k,
b.ag_k AS ag_k,
b.kgaa_k AS kgaa_k,
b.eg_k AS eg_k,
b.verein_k AS verein_k,
b.vvag_k AS vvag_k,
b.gmbh_k AS gmbh_k,
b.sgs_k AS sgs_k,
b.firma_k AS firma_k,
b.liq_k AS liq_k,
b.mj_k AS mj_k,
b.gu_k AS gu_k,
b.w_k AS w_k,
b.getr_k AS getr_k,
b.eheleute_k AS eheleute_k,
b.gg_k AS gg_k,
b.gt_k AS gt_k,
b.eaez_k AS eaez_k,
b.done_k AS done_k,
b.bemerk AS bemerk,
b.`LAST` AS `LAST`,
b.ts AS ts
FROM
(((k_beend b
LEFT JOIN ngericht ngg ON
((b.gericht = ngg.gericht))
LEFT JOIN tgerichtort ON
((COALESCE(ngg.gerichtn,
b.gericht) = tgerichtort.gerichtn))
LEFT JOIN nregiogericht ON
((COALESCE(ngg.gerichtn,
b.gericht) = nregiogericht.gerichtn))
LEFT JOIN nort ON
((b.s_ort = nort.ort))) ;
-- Struktur des Views k_beend_cnp
CREATE
ALGORITHM = MERGE DEFINER = reichsanzeiger@localhost SQL SECURITY INVOKER VIEW
k_beend_cnp AS SELECT
b.`id` AS `id`,
b.aid AS aid,
b.pid AS pid,
b.kid AS kid,
b.s_name AS s_name,
b.s_beruf AS s_beruf,
b.s_ort AS s_ort,
b.gericht AS gericht,
b.gericht_cn AS gericht_cn,
b.gerichtort_cn AS gerichtort_cn,
b.regiogericht_cn AS regiogericht_cn,
b.typ AS typ,
b.aufh_dat AS aufh_dat,
b.aufh_dat_c AS aufh_dat_c,
b.bek_dat AS bek_dat,
b.t_dat AS t_dat,

```

```

b.flucht_k AS flucht_k,
b.haft_k AS haft_k,
b.abwesend_k AS abwesend_k,
b.unbaufent_k AS unbaufent_k,
b.nachlass_k AS nachlass_k,
b.ohg_k AS ohg_k,
b.kg_k AS kg_k,
b.gesellter_k AS gesellter_k,
b.ag_k AS ag_k,
b.kgaa_k AS kgaa_k,
b.eg_k AS eg_k,
b.verein_k AS verein_k,
b.vvag_k AS vvag_k,
b.gmbh_k AS gmbh_k,
b.sgs_k AS sgs_k,
b.firma_k AS firma_k,
b.liq_k AS liq_k,
b.mj_k AS mj_k,
b.gu_k AS gu_k,
b.w_k AS w_k,
b.getr_k AS getr_k,
b.eheleute_k AS eheleute_k,
b.gg_k AS gg_k,
b.gt_k AS gt_k,
b.eaez_k AS eaez_k,
b.done_k AS done_k,
b.bemerk AS bemerk,
b.`LAST` AS `LAST`,
b.ts AS ts
FROM
  k_beend_cn b
WHERE
  ((b.eaez_k = 0)
  AND (NOT((b.bemerk LIKE '%#IGN%'))))
  AND (NOT((b.bemerk LIKE '%#ELV%'))))
  AND (NOT(b.`id` IN (
    SELECT k_beend.pid
  FROM
    k_beend
  WHERE
    (k_beend.pid IS NOT NULL)))))) ;

```

```

-- Struktur des Views k_eroef_cn
CREATE
  ALGORITHM = MERGE DEFINER = reichsanzeiger@localhost SQL SECURITY INVOKER VIEW
k_eroef_cn AS SELECT
  k.`id` AS `id`,
  k.aid AS aid,
  k.pid AS pid,
  k.s_name AS s_name,
  k.s_beruf AS s_beruf,
  k.s_ort AS s_ort,
  k.v_name AS v_name,
  k.v_beruf AS v_beruf,
  COALESCE(nb.berufn,
  k.v_beruf) AS v_beruf_cn,
  k.v_ort AS v_ort,
  COALESCE(nvo.ortn,
  COALESCE(tgvo.ortn,
  COALESCE(ngvo.gerichtn,
  k.v_ort))) AS v_ort_cn,
  k.vid AS vid,
  k.eroef_dat AS eroef_dat,
  IF((k.eroef_dat = '0000-00-00'),
  k.bek_dat,
  k.eroef_dat) AS eroef_dat_c,
  k.bek_dat AS bek_dat,
  k.anz_dat AS anz_dat,
  k.anm_dat AS anm_dat,
  k.gvers dat AS gvers dat,

```

```

k.pruef_dat AS pruef_dat,
k.gericht AS gericht,
COALESCE(ngg.gerichtn,
k.gericht) AS gericht_cn,
COALESCE(tgg.ortn,
COALESCE(ngg.gerichtn,
k.gericht)) AS gerichtort_cn,
COALESCE(nregiogericht.regiogericht,
COALESCE(ngg.gerichtn,
k.gericht)) AS regiogericht_cn,
k.t_dat AS t_dat,
k.flucht_k AS flucht_k,
k.haft_k AS haft_k,
k.abwesend_k AS abwesend_k,
k.unbaufent_k AS unbaufent_k,
k.nachlass_k AS nachlass_k,
k.ohg_k AS ohg_k,
k.kg_k AS kg_k,
k.gesellter_k AS gesellter_k,
k.ag_k AS ag_k,
k.kgaa_k AS kgaa_k,
k.eg_k AS eg_k,
k.verein_k AS verein_k,
k.vvag_k AS vvag_k,
k.gmbh_k AS gmbh_k,
k.sgs_k AS sgs_k,
k.firma_k AS firma_k,
k.liq_k AS liq_k,
k.mj_k AS mj_k,
k.gu_k AS gu_k,
k.w_k AS w_k,
k.getr_k AS getr_k,
k.eheleute_k AS eheleute_k,
k.gg_k AS gg_k,
k.gt_k AS gt_k,
k.done_k AS done_k,
k.bemerk AS bemerk,
k.`LAST` AS `LAST`,
k.ts AS ts
FROM
(((((((k_eroef k
LEFT JOIN nberuf nb ON
((k.v_beruf = nb.beruf)))
LEFT JOIN ngericht ngg ON
((k.gericht = ngg.gericht)))
LEFT JOIN nregiogericht ON
((COALESCE(ngg.gerichtn,
k.gericht) = nregiogericht.gerichtn)))
LEFT JOIN tgerichtort tgg ON
((COALESCE(ngg.gerichtn,
k.gericht) = tgg.gerichtn)))
LEFT JOIN ngericht ngvo ON
((k.v_ort = ngvo.gericht)))
LEFT JOIN tgerichtort tgvo ON
((COALESCE(ngvo.gerichtn,
k.v_ort) = tgvo.gerichtn)))
LEFT JOIN nort nvo ON
((COALESCE(tgvo.ortn,
COALESCE(ngvo.gerichtn,
k.v_ort)) = nvo.ort))) ;
-- Struktur des Views k_eroef_cnp
CREATE
ALGORITHM = MERGE DEFINER = reichsanzeiger@localhost SQL SECURITY INVOKER VIEW
k_eroef_cnp AS SELECT
e.`id` AS `id`,
e.aid AS aid,
e.pid AS pid,
e.s_name AS s_name,
e.s_beruf AS s_beruf,

```



```

e.s_ort AS s_ort,
e.v_name AS v_name,
e.v_beruf AS v_beruf,
e.v_beruf_cn AS v_beruf_cn,
e.v_ort AS v_ort,
e.v_ort_cn AS v_ort_cn,
e.vid AS vid,
e.eroef_dat AS eroef_dat,
e.eroef_dat_c AS eroef_dat_c,
e.bek_dat AS bek_dat,
e.anz_dat AS anz_dat,
e.anm_dat AS anm_dat,
e.gvers_dat AS gvers_dat,
e.pruef_dat AS pruef_dat,
e.gericht AS gericht,
e.gericht_cn AS gericht_cn,
e.gerichtort_cn AS gerichtort_cn,
e.regiogericht_cn AS regiogericht_cn,
e.t_dat AS t_dat,
e.flucht_k AS flucht_k,
e.haft_k AS haft_k,
e.abwesend_k AS abwesend_k,
e.unbaufent_k AS unbaufent_k,
e.nachlass_k AS nachlass_k,
e.ohg_k AS ohg_k,
e.kg_k AS kg_k,
e.gesellter_k AS gesellter_k,
e.ag_k AS ag_k,
e.kgaa_k AS kgaa_k,
e.eg_k AS eg_k,
e.verein_k AS verein_k,
e.vvag_k AS vvag_k,
e.gmbh_k AS gmbh_k,
e.sgs_k AS sgs_k,
e.firma_k AS firma_k,
e.liq_k AS liq_k,
e.mj_k AS mj_k,
e.gu_k AS gu_k,
e.w_k AS w_k,
e.getr_k AS getr_k,
e.eheleute_k AS eheleute_k,
e.gg_k AS gg_k,
e.gt_k AS gt_k,
e.done_k AS done_k,
e.bemerk AS bemerk,
e.`LAST` AS `LAST`,
e.ts AS ts

```

```

FROM
k_eroef_cn e
WHERE
((NOT((e.bemerk LIKE '%#IGN%'))))
AND (NOT((e.bemerk LIKE '%#ELV%'))))
AND (NOT(e.`id` IN (
SELECT k_eroef.pid
FROM
k_eroef
WHERE
(k_eroef.pid IS NOT NULL)))));

```

```

-- Struktur des Views vid_c
CREATE
ALGORITHM = UNDEFINED DEFINER = reichsanzeiger@localhost SQL SECURITY INVOKER
VIEW vid_c AS SELECT
e.vid AS vid,
e.v_beruf_cn AS v_beruf_cn,
COUNT(0) AS num,
i.maxnum AS maxnum
FROM
(k_eroef_cnp e
LEFT JOIN vid_maxnum i ON

```

```

        ((e.vid = i.vid)))
    GROUP BY
        e.vid,
        e.v_beruf_cn ;

```

---

```

-- Struktur des Views vid_cm
CREATE
    ALGORITHM = UNDEFINED DEFINER = root@localhost SQL SECURITY DEFINER VIEW vid_cm
AS SELECT
    vid_c.vid AS vid,
    MIN(NULLIF(vid_c.v_beruf_cn, '')) AS v_beruf_cn
FROM
    vid_c
WHERE
    ((vid_c.num = vid_c.maxnum)
    OR isnull(vid_c.maxnum))
GROUP BY
    vid_c.vid ;

```

---

```

-- Struktur des Views vid_maxnum
CREATE
    ALGORITHM = UNDEFINED DEFINER = reichsanzeiger@localhost SQL SECURITY INVOKER
VIEW vid_maxnum AS SELECT
    vn1.vid AS vid,
    MAX(vn1.num) AS maxnum
FROM
    vid_num vn1
WHERE
    ((vn1.vid IS NOT NULL)
    AND ((vn1.v_beruf_cn <> '')
    OR (NOT (EXISTS(
        SELECT 1
        FROM
            vid_num vn2
        WHERE
            ((vn2.vid = vn1.vid)
            AND (vn2.v_beruf_cn <> '')))))))
GROUP BY
    vn1.vid ;

```

---

```

-- Struktur des Views vid_num
CREATE
    ALGORITHM = UNDEFINED DEFINER = reichsanzeiger@localhost SQL SECURITY INVOKER
VIEW vid_num AS SELECT
    e.vid AS vid,
    e.v_beruf_cn AS v_beruf_cn,
    COUNT(0) AS num
FROM
    k_eroef_cnp e
GROUP BY
    e.vid,
    e.v_beruf_cn ;

```

#### d. Indizes

```

-- Indizes für die Tabelle k_eroef
ALTER TABLE
    k_eroef ADD PRIMARY KEY (id),
    ADD KEY aid (aid),
    ADD KEY gericht (gericht),
    ADD KEY v_name (v_name(767)),
    ADD KEY vid (vid),
    ADD KEY v_ort (v_ort),
    ADD KEY v_beruf (v_beruf),
    ADD KEY s_ort (s_ort),
    ADD KEY s_name (s_name(767)),
    ADD KEY s_beruf (s_beruf),
    ADD KEY pid (pid)
    USING BTREE;
-- Indizes für die Tabelle k_beend

```

```

ALTER TABLE
  k_beend ADD PRIMARY KEY (id),
  ADD UNIQUE KEY kid (kid)
  USING BTREE,
  ADD KEY aid (aid,
  kid),
  ADD KEY s_ort (s_ort),
  ADD KEY gericht (gericht),
  ADD KEY eaez (eaez_k),
  ADD KEY s_name (s_name(767)),
  ADD KEY s_beruf (s_beruf),
  ADD KEY pid (pid)
  USING BTREE;

-- Indizes für die Tabelle nberuf
ALTER TABLE
  nberuf ADD PRIMARY KEY (id),
  ADD UNIQUE KEY beruf (beruf);

-- Indizes für die Tabelle ngericht
ALTER TABLE
  ngericht ADD PRIMARY KEY (id),
  ADD UNIQUE KEY gericht (gericht),
  ADD KEY gerichtn (gerichtn);

-- Indizes für die Tabelle nort
ALTER TABLE
  nort ADD PRIMARY KEY (id),
  ADD UNIQUE KEY ort (ort);

-- Indizes für die Tabelle nregiogericht
ALTER TABLE
  nregiogericht ADD PRIMARY KEY (id),
  ADD UNIQUE KEY gerichtn (gerichtn),
  ADD KEY regiogericht (regiogericht);

-- Indizes für die Tabelle tgerichtort
ALTER TABLE
  tgerichtort ADD PRIMARY KEY (id),
  ADD UNIQUE KEY gerichtn (gerichtn),
  ADD KEY ortn (ortn);

-- Indizes für die Tabelle titelbl
ALTER TABLE
  titelbl ADD PRIMARY KEY (id),
  ADD UNIQUE KEY fid_bildnr (bildnr,
  fid),
  ADD KEY fid (fid);

-- Indizes für die Tabelle verw
ALTER TABLE
  verw ADD PRIMARY KEY (id),
  ADD KEY berufnm (berufnm);

-- Indizes für die Tabelle `filme`
ALTER TABLE
  `filme` ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `filmnr_combined` (`filmnr_kurz`,
  `filmnr_lang`);

```

#### e. Fremdschlüsselbeschränkungen

```

-- Constraints der Tabelle k_beend
ALTER TABLE
  k_beend ADD CONSTRAINT k_beend_ibfk_1 FOREIGN KEY (kid) REFERENCES k_eroef
  (id),
  ADD CONSTRAINT k_beend_ibfk_2 FOREIGN KEY (aid) REFERENCES kmit (id),
  ADD CONSTRAINT k_beend_ibfk_3 FOREIGN KEY (pid) REFERENCES k_beend (id);

-- Constraints der Tabelle k_eroef
ALTER TABLE
  k_eroef ADD CONSTRAINT k_eroef_ibfk_1 FOREIGN KEY (aid) REFERENCES kmit (id),
  ADD CONSTRAINT k_eroef_ibfk_2 FOREIGN KEY (vid) REFERENCES verw (id) ON
  DELETE
  SET
  NULL ON
  UPDATE
  CASCADE,

```

<pre> ADD CONSTRAINT k_eroef_ibfk_3 FOREIGN KEY (pid) REFERENCES k_eroef (id); -- Constraints der Tabelle titelbl ALTER TABLE titelbl ADD CONSTRAINT titelbl_ibfk_1 FOREIGN KEY (fid) REFERENCES filme (id); </pre>
---

## 12. Liste der Datenbank-Abfragen (SQL)

Q1	<pre> SELECT COUNT(*) FROM k_eroef_cnp; </pre>
Q2	<pre> SELECT COUNT(*) FROM k_beend_cnp; </pre>
Q3	<pre> SELECT YEAR(E.eroef_dat_c), COUNT(E.id), COUNT(B.id), COUNT(B.id)/ COUNT(E.id) FROM `k_eroef_cnp` E LEFT JOIN k_beend_cnp B ON e.id = b.kid GROUP BY YEAR(E.eroef_dat_c) ORDER BY YEAR(E.eroef_dat_c); </pre>
Q4	<pre> SELECT YEAR(B.aufh_dat_c), COUNT(B.id), COUNT(E.id), COUNT(E.id)/ COUNT(B.id) FROM k_beend_cnp B LEFT JOIN `k_eroef_cnp` E ON e.id = b.kid GROUP BY YEAR(B.aufh_dat_c) ORDER BY YEAR(B.aufh_dat_c); </pre>
Q5	<pre> SELECT COUNT(*) FROM `k_eroef` e WHERE E.bemerck NOT LIKE "%#IGN%" AND E.bemerck NOT LIKE "%#ELV%" AND e.pid IS NULL; </pre>
Q6	<pre> SELECT COUNT(*) FROM `k_beend` b WHERE B.`eaez_k` = 0 AND b.bemerck NOT LIKE "%#IGN%" AND b.bemerck NOT LIKE "%#ELV%" AND b.pid IS NULL; </pre>
Q7	<pre> SET @a = "s_name"; -- für alle Zeichenkettenfelder SET @query = CONCAT("SELECT '", @a, "', (SELECT count(*) FROM `k_eroef_cnp` WHERE ", @a, " != '') as num, (SELECT count(*) FROM `k_eroef_cnp`) as gesamt, (SELECT count(*) FROM `k_eroef_cnp` WHERE ", @a, " != '') / (SELECT count(*) FROM `k_eroef_cnp`) as quote"); PREPARE stmt FROM @query; EXECUTE stmt; </pre>
Q8	<pre> SET @a = 'eroef_dat'; -- für alle Datumsfelder der Eröffnung SET @query = CONCAT("SELECT '", @a, "', (SELECT count(*) FROM `k_eroef_cnp` WHERE ", @a, " != '0000-00-00') as num, (SELECT count(*) FROM `k_eroef_cnp`) as gesamt, (SELECT count(*) FROM `k_eroef_cnp` WHERE ", @a, " != '0000-00-00') / (SELECT count(*) FROM `k_eroef_cnp`) as quote"); PREPARE stmt FROM @query; EXECUTE stmt; </pre>
Q9	<pre> SET @a = 'aufh_dat'; -- für alle Datumsfelder der Beendigung SET @query = CONCAT("SELECT '", @a, "', (SELECT count(*) FROM `k_beend_cnp` WHERE ", @a, " != '0000-00-00') as num, (SELECT count(*) FROM `k_beend_cnp`) as gesamt, (SELECT count(*) FROM `k_beend_cnp` WHERE ", @a, " != '0000-00-00') / (SELECT count(*) FROM `k_beend_cnp`) as quote"); PREPARE stmt FROM @query; EXECUTE stmt; </pre>
Q10	<pre> SET </pre>

	<pre> @a = 'typ'; SET @query = CONCAT("SELECT '", @a, "', (SELECT count(*) FROM `k_beend_cnp` WHERE ", @a, " != 'u') as num, (SELECT count(*) FROM `k_beend_cnp`) as gesamt, (SELECT count(*) FROM `k_beend_cnp` WHERE ", @a, " != 'u') / (SELECT count(*) FROM `k_beend_cnp`) as quote"); PREPARE stmt FROM @query; EXECUTE stmt; </pre>
Q11	<pre> SELECT COUNT(b.gericht)/ COUNT(*) FROM k_beend_cnp b JOIN k_eroef_cnp e ON b.kid = e.id; </pre>
Q12	<pre> SELECT COUNT(b.s_name)/ COUNT(*) FROM k_beend_cnp b JOIN k_eroef_cnp e ON b.kid = e.id; </pre>
Q13	<pre> SELECT COUNT(b.s_ort)/ COUNT(*) FROM k_beend_cnp b JOIN k_eroef_cnp e ON b.kid = e.id; </pre>
Q14	<pre> SELECT COUNT(b.s_beruf)/ COUNT(*) FROM k_beend_cnp b JOIN k_eroef_cnp e ON b.kid = e.id; </pre>
Q15	<pre> SELECT SUM(b.t_dat != "0000-00-00")/ COUNT(*) FROM k_beend_cnp b JOIN k_eroef_cnp e ON b.kid = e.id; </pre>
Q16	<pre> SELECT COUNT(b.gericht)/ COUNT(*) FROM k_beend_cnp b WHERE kid IS NULL; </pre>
Q17	<pre> SELECT COUNT(b.s_name)/ COUNT(*) FROM k_beend_cnp b WHERE kid IS NULL; </pre>
Q18	<pre> SELECT COUNT(b.s_ort)/ COUNT(*) FROM k_beend_cnp b WHERE kid IS NULL; </pre>
Q19	<pre> SELECT COUNT(b.s_beruf)/ COUNT(*) FROM k_beend_cnp b WHERE kid IS NULL; </pre>
Q20	<pre> SELECT SUM(t_dat != "0000-00-00")/ COUNT(*) FROM k_beend_cnp b WHERE kid IS NULL; </pre>
Q21	<pre> SELECT COUNT(b.id), COUNT(b.id)/ COUNT(*) FROM k_eroef_cnp e LEFT JOIN k_beend_cnp b ON e.id = b.kid; </pre>
Q22	<pre> SELECT COUNT(e.id), COUNT(e.id)/ COUNT(*) FROM k_beend_cnp b LEFT JOIN k_eroef_cnp e ON b.kid = e.id; </pre>
Q23	<pre> SELECT COUNT(*), e.gericht_cn, e.v_ort_cn, e.vid FROM k_eroef_cnp e WHERE e.gerichtort_cn != e.v_ort_cn AND e.v_ort_cn != "" GROUP BY e.gericht_cn, e.v_ort_cn, e.vid ORDER BY e.gericht_cn, e.v_ort_cn, e.vid; </pre>
Q24	<pre> SELECT * FROM k_eroef_cnp WHERE vid = 781; </pre>
Q25	<pre> SELECT COUNT(DISTINCT vid) FROM `k_eroef_cnp` WHERE vid IS NOT NULL </pre>

	<b>AND vid != 0;</b>
Q26	<b>SELECT COUNT(*) FROM k_eroef_cnp e WHERE e.gerichtort_cn != e.v_ort_cn AND e.v_ort_cn != "" AND e.id NOT IN (35916, 25918, 35933, 10424, 49242, 49316, 49779, 52702, 50766, 52788, 43455, 31403, 30755, 50947, 50922, 2516);</b>
Q27	<b>SELECT * FROM k_eroef_cnp e LEFT JOIN k_beend_cnp b ON e.id = b.kid UNION SELECT * FROM k_eroef_cnp e RIGHT JOIN k_beend_cnp b ON e.id = b.kid;</b>
Q28	<b>SELECT gericht_cn, COUNT(*) FROM k_eroef_cnp e GROUP BY gericht_cn ORDER BY gericht_cn;</b>
Q29	<b>SELECT COALESCE(b.gericht_cn, e.gericht_cn), COUNT(*) FROM k_beend_cnp b LEFT JOIN k_eroef_cnp e ON b.kid = e.id GROUP BY COALESCE(b.gericht_cn, e.gericht_cn) ORDER BY COALESCE(b.gericht_cn, e.gericht_cn);</b>
Q30	<b>SELECT regiogericht_cn, COUNT(*) FROM k_eroef_cnp e GROUP BY regiogericht_cn ORDER BY regiogericht_cn;</b>
Q31	<b>SELECT COALESCE(b.regiogericht_cn, e.regiogericht_cn), COUNT(*) FROM k_beend_cnp b LEFT JOIN k_eroef_cnp e ON b.kid = e.id GROUP BY COALESCE(b.regiogericht_cn, e.regiogericht_cn) ORDER BY COALESCE(b.regiogericht_cn, e.regiogericht_cn);</b>
Q32	<b>SELECT COUNT(*) FROM k_eroef_cnp WHERE v_ort_cn = "Blasewitz" AND gericht_cn = "Dresden";</b>
Q33	<b>SELECT COUNT(*), regiogericht_cn FROM k_eroef_cnp e WHERE e.gerichtort_cn != e.v_ort_cn AND e.v_ort_cn != "" AND e.id NOT IN (35916, 25918, 35933, 10424, 49242, 49316, 49779, 52702, 50766, 52788, 43455, 31403, 30755, 50947, 50922, 2516) GROUP BY regiogericht_cn ORDER BY regiogericht_cn;</b>
Q34	<b>SELECT COUNT(*), regiogericht_cn FROM k_eroef_cnp e WHERE e.gerichtort_cn != e.v_ort_cn AND e.v_ort_cn != "" AND e.id NOT IN (35916, 25918, 35933, 10424, 49242, 49316, 49779, 52702, 50766, 52788, 43455, 31403, 30755, 50947, 50922, 2516) AND e.gericht_cn NOT IN ("Königsberg i. Pr.", "Mannheim", "Straßburg i. Pr.") GROUP BY regiogericht_cn ORDER BY regiogericht_cn;</b>
Q35	<b>SELECT COUNT(*), regiogericht_cn FROM k_eroef_cnp e WHERE e.gerichtort_cn != e.v_ort_cn AND e.v_ort_cn != "" AND e.id NOT IN (35916, 25918, 35933, 10424, 49242, 49316, 49779, 52702, 50766, 52788, 43455, 31403, 30755, 50947, 50922, 2516) AND e.gericht_cn IN ("Königsberg i. Pr.", "Mannheim", "Straßburg i. Pr.") GROUP BY regiogericht_cn ORDER BY regiogericht_cn;</b>

Q36	<pre> SELECT COUNT(*) AS gesamt, SUM(e.gerichtort_cn != e.v_ort_cn AND e.v_ort_cn != "") AS ausserhalb FROM k_eroef_cnp e WHERE e.id NOT IN (35916, 25918, 35933, 10424, 49242, 49316, 49779, 52702, 50766, 52788, 43455, 31403, 30755, 50947, 50922, 2516) AND (regiogericht_cn NOT IN ("LG Königsberg", "LG Mannheim", "LG Straßburg") OR gericht_cn IN ("Königsberg i. Pr.", "Mannheim", "Straßburg i. Pr.)); </pre>
Q37	<pre> SELECT COUNT(*), e.v_ort_cn FROM k_eroef_cnp e WHERE e.gerichtort_cn != e.v_ort_cn AND e.v_ort_cn != "" AND e.regiogericht_cn IN ("LG Königsberg", "LG Mannheim", "LG Straßburg") GROUP BY e.v_ort_cn ORDER BY e.v_ort_cn; </pre>
Q38	<pre> SELECT COUNT(*), e.regiogericht_cn FROM k_eroef_cnp e WHERE e.gerichtort_cn != e.v_ort_cn AND e.v_ort_cn != "" AND e.regiogericht_cn IN ("LG Königsberg", "LG Mannheim", "LG Straßburg") AND e.id NOT IN (35916, 25918, 35933, 10424, 49242, 49316, 49779, 52702, 50766, 52788, 43455, 31403, 30755, 50947, 50922, 2516) AND (e.v_ort_cn = e.s_ort OR replace(e.s_ort, e.v_ort_cn, "") != e.s_ort) GROUP BY e.regiogericht_cn ORDER BY e.regiogericht_cn; </pre>
Q39	<pre> SELECT COUNT(DISTINCT v_beruf) FROM `k_eroef_cnp` ; </pre>
Q40	<pre> SELECT v_beruf_cn, v_beruf, COUNT(*) FROM `k_eroef_cnp` GROUP BY v_beruf ORDER BY v_beruf; </pre>
Q41	<pre> SELECT v_beruf_cn, (vid IS NULL), COUNT(*) FROM `k_eroef_cnp` GROUP BY v_beruf_cn, (vid IS NULL) ORDER BY v_beruf_cn; </pre>
Q42	<pre> SELECT SUM(num) FROM ( SELECT vid, COUNT(*) AS num FROM k_eroef_cnp WHERE vid IS NOT NULL GROUP BY vid HAVING COUNT(DISTINCT v_beruf_cn) &gt; 1) t; </pre>
Q43	<pre> SELECT COUNT(DISTINCT vid) FROM ( SELECT vid, COUNT(*) AS num FROM k_eroef_cnp GROUP BY vid HAVING COUNT(DISTINCT v_beruf_cn) &gt; 1) t; </pre>
Q44	<pre> SELECT COUNT(*) / COUNT(DISTINCT vid) FROM k_eroef_cnp WHERE vid IS NOT NULL; </pre>
Q45	<pre> SELECT vid, SUM(num), maxnum, maxnum / SUM(num) AS q, COUNT(*) AS berufe FROM ( SELECT e.vid, e.v_beruf_cn, COUNT(*) AS num, i.maxnum FROM k_eroef_cnp e INNER JOIN ( SELECT vid, MAX(num) AS maxnum FROM ( SELECT vid, COUNT(*) AS num FROM k_eroef_cnp e GROUP BY vid, v_beruf_cn) j GROUP BY vid ) i ON e.vid = i.vid GROUP BY e.vid, e.v_beruf_cn ORDER BY e.vid, e.v_beruf_cn) t GROUP BY vid HAVING COUNT(*) &gt; 1 ORDER BY q DESC; </pre>
Q46	<pre> SELECT vid, v_beruf_cn, COUNT(*), e.* FROM k_eroef_cnp e WHERE vid = 2 </pre>

	<b>GROUP BY v_beruf_cn;</b>
Q47	<pre> SELECT vid, SUM(num), maxnum, maxnum / SUM(num) AS q, COUNT(*) AS berufe FROM ( SELECT e.vid, e.v_beruf_cn, COUNT(*) AS num, i.maxnum FROM k_eroef_cnp e INNER JOIN ( SELECT vid, MAX(num) AS maxnum FROM ( SELECT vid, COUNT(*) AS num FROM k_eroef_cnp e GROUP BY vid, v_beruf_cn) j GROUP BY vid ) i ON e.vid = i.vid GROUP BY e.vid, e.v_beruf_cn ORDER BY e.vid, e.v_beruf_cn) t GROUP BY vid HAVING COUNT(*) &gt; 1 AND maxnum / SUM(num) &gt; 0.95 ORDER BY q DESC; </pre>
Q48	<pre> SELECT SUM(num) FROM ( SELECT vid, SUM(num) AS num, maxnum, maxnum / SUM(num) AS q, COUNT(*) AS berufe FROM ( SELECT e.vid, e.v_beruf_cn, COUNT(*) AS num, i.maxnum FROM k_eroef_cnp e INNER JOIN ( SELECT vid, MAX(num) AS maxnum FROM ( SELECT vid, COUNT(*) AS num FROM k_eroef_cnp e GROUP BY vid, v_beruf_cn) j GROUP BY vid ) i ON e.vid = i.vid GROUP BY e.vid, e.v_beruf_cn ORDER BY e.vid, e.v_beruf_cn) t GROUP BY vid HAVING COUNT(*) &gt; 1 AND maxnum / SUM(num) &gt; 0.95 ORDER BY q DESC) x; </pre>
Q49	<pre> SELECT SUM(berufe - 1) FROM ( SELECT vid, SUM(num) AS num, maxnum, maxnum / SUM(num) AS q, COUNT(*) AS berufe FROM ( SELECT e.vid, e.v_beruf_cn, COUNT(*) AS num, i.maxnum FROM k_eroef_cnp e INNER JOIN ( SELECT vid, MAX(num) AS maxnum FROM ( SELECT vid, COUNT(*) AS num FROM k_eroef_cnp e GROUP BY vid, v_beruf_cn) j GROUP BY vid ) i ON e.vid = i.vid GROUP BY e.vid, e.v_beruf_cn ORDER BY e.vid, e.v_beruf_cn) t GROUP BY vid HAVING COUNT(*) &gt; 1 ORDER BY q DESC ) x; </pre>
Q50	<pre> SELECT COUNT(*) FROM `nberuf`; </pre>
Q51	<pre> SELECT * FROM vid_c WHERE vid IN ( SELECT vid FROM `vid_c` WHERE num = maxnum GROUP BY vid HAVING COUNT(*) &gt; 1); </pre>
Q52	<pre> SELECT v_beruf_cn, COUNT(*) FROM `vid_cm` WHERE vid IS NOT NULL GROUP BY v_beruf_cn ORDER BY v_beruf_cn IS NULL, v_beruf_cn; </pre>
Q53	<pre> SELECT COALESCE(v.v_beruf_cn, e.v_beruf_cn), (e.vid IS NULL), COUNT(*) FROM k_eroef_cnp e LEFT JOIN vid_cm v ON v.vid = e.vid </pre>



	<b>GROUP BY COALESCE(v.v_beruf_cn, e.v_beruf_cn), (e.vid IS NULL) ORDER BY COALESCE(v.v_beruf_cn, e.v_beruf_cn);</b>
Q54	<b>SELECT v_beruf_cn, COUNT(*) FROM `vid_c` WHERE vid IS NOT NULL GROUP BY v_beruf_cn ORDER BY v_beruf_cn = '', v_beruf_cn;</b>
Q55	<b>SELECT * FROM vid_num WHERE vid IN ( SELECT vid FROM vid_num WHERE (vid, num) IN ( SELECT vid, MAX(num) AS maxnum FROM vid_num GROUP BY vid) AND v_beruf_cn = "');</b>
Q56	<b>SELECT e.vid, v.v_beruf_cn, e.v_beruf_cn, COUNT(*) FROM k_eroef_cnp e JOIN vid_num v ON e.vid = v.vid WHERE e.gericht_cn = "Hamburg" AND e.vid IN ( SELECT DISTINCT vid FROM k_eroef_cnp WHERE gericht_cn = "Hamburg" AND v_beruf_cn = "Bücherrevisor") GROUP BY e.vid, e.v_beruf_cn;</b>
Q57	<b>SELECT * FROM `titelbl` WHERE (jahr &gt; 1879 AND jahr &lt; 1914) OR (jahr = 1879 AND ausgabe &gt;= 230) OR (jahr = 1914 AND ausgabe &lt;= 178) GROUP BY zeitung, ausgabe, jahr;</b>
Q58	<b>SELECT * FROM `titelbl` GROUP BY zeitung, ausgabe, jahr;</b>
Q59	<b>SELECT * FROM `titelbl` WHERE (jahr &gt; 1879) OR (jahr = 1879 AND ausgabe &gt;= 230) GROUP BY zeitung, ausgabe, jahr;</b>
Q60	<b>SELECT COUNT(*) FROM `k_eroef_cnp` WHERE gericht_cn IN ("Berlin I", "Breslau", "Chemnitz", "Dresden", "Düsseldorf", "Frankfurt a. M.", "Hamburg", "Hannover", "Köln", "Königsberg i. Pr.", "Magdeburg", "München I", "Nürnberg", "Leipzig", "Stuttgart, Stadt") ORDER BY `vid` ASC;</b>
Q61	<b>SELECT COUNT(*) FROM `k_eroef_cnp` WHERE gericht_cn IN ("Berlin I", "Breslau", "Chemnitz", "Dresden", "Düsseldorf", "Frankfurt a. M.", "Hamburg", "Hannover", "Köln", "Königsberg i. Pr.", "Magdeburg", "München I", "Nürnberg", "Leipzig", "Stuttgart, Stadt") AND YEAR(eroef_dat_c) = 1896 ORDER BY `vid` ASC;</b>
Q62	<b>SELECT COUNT(*) FROM `k_eroef_cnp` WHERE gericht_cn NOT IN ("Berlin I", "Breslau", "Chemnitz", "Dresden", "Düsseldorf", "Frankfurt a. M.", "Hamburg", "Hannover", "Köln", "Königsberg i. Pr.", "Magdeburg", "München I", "Nürnberg", "Leipzig", "Stuttgart, Stadt") AND YEAR(eroef_dat_c) = 1896 ORDER BY `vid` ASC;</b>
Q63	<b>SELECT YEAR(eroef_dat_c), COUNT(*) FROM `k_eroef_cnp` WHERE YEAR(eroef_dat_c) &gt;= 1880</b>

	<pre> AND YEAR(eroef_dat_c) &lt;= 1913 GROUP BY YEAR(eroef_dat_c) ORDER BY YEAR(eroef_dat_c); </pre>
Q64	<pre> SELECT YEAR(eroef_dat_c), SUM(gericht_cn = "Berlin I"), SUM(gericht_cn = "Dresden"), SUM(gericht_cn = "Leipzig"), SUM(gericht_cn = "Stuttgart, Stadt") FROM `k_eroef_cnp` WHERE YEAR(eroef_dat_c) &gt;= 1895 AND YEAR(eroef_dat_c) &lt;= 1913 GROUP BY YEAR(eroef_dat_c) ORDER BY YEAR(eroef_dat_c); </pre>
Q65	<pre> SELECT * FROM `k_eroef_cnp` WHERE YEAR(eroef_dat_c) = 1896 AND (ag_k = 1) AND gericht_cn IN ("Berlin I", "Breslau", "Chemnitz", "Dresden", "Düsseldorf", "Frankfurt a. M.", "Hamburg", "Hannover", "Köln", "Königsberg i. Pr.", "Leipzig", "Magdeburg", "München I", "Nürnberg", "Stuttgart, Stadt"); </pre>
Q66	<pre> SELECT regiogericht_cn, YEAR(eroef_dat_c), COUNT(*) FROM k_eroef_cnp GROUP BY regiogericht_cn, YEAR(eroef_dat_c) ORDER BY regiogericht_cn, YEAR(eroef_dat_c); </pre>
Q67	<pre> SELECT YEAR(E.eroef_dat_c), COUNT(*) FROM `k_eroef_cn` E WHERE E.regiogericht_cn = "Bremen" AND E.bemerk NOT LIKE "%#IGN%" AND E.id NOT IN ( SELECT pid FROM k_eroef WHERE pid IS NOT NULL) GROUP BY YEAR(E.eroef_dat_c) ORDER BY YEAR(E.eroef_dat_c); </pre>
Q68	<pre> SELECT e.id, gericht_cn, vid, v_ort FROM `k_eroef_cnp` e WHERE e.id IN (35916, 25918, 35933, 10424, 49242, 49316, 49779, 52702, 50766, 52788, 43455, 31403, 30755, 50947, 50922, 2516) ORDER BY vid; </pre>
Q69	<pre> SELECT COUNT(e.id), COUNT(b.id), COUNT(b.id) / COUNT(e.id) FROM `k_eroef_cnp` E LEFT JOIN k_beend_cnp B ON e.id = b.kid WHERE YEAR(E.eroef_dat_c) &lt;= 1908; </pre>
Q70	<pre> SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT SUM(firma_k) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t; </pre>
Q71	<pre> SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT SUM(eroef_dat_c != "0000-00-00") AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t; </pre>
Q72	<pre> SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT COUNT(DATEDIFF(bek_dat, eroef_dat_c)) - SUM(datediff(bek_dat, eroef_dat_c) &lt; 0) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t; </pre>
Q73	<pre> SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT COUNT(DATEDIFF(gvers_dat, eroef_dat_c)) - SUM(DATEDIFF(gvers_dat, eroef_dat_c) &lt; 0) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t; </pre>
Q74	<pre> SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT COUNT(DATEDIFF(anz_dat, eroef_dat_c)) - SUM(DATEDIFF(anz_dat, eroef_dat_c) &lt; 0) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t; </pre>
Q75	<pre> SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT COUNT(DATEDIFF(anm_dat, eroef_dat_c)) - SUM(DATEDIFF(anm_dat, eroef_dat_c) &lt; 0) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t; </pre>
Q76	<pre> SELECT vorhanden, gesamt, vorhanden / gesamt AS quote </pre>

	<b>FROM ( SELECT COUNT(DATEDIFF(pruef_dat, eroef_dat_c))-SUM(DATEDIFF(pruef_dat, eroef_dat_c) &lt; 0) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t;</b>
Q77	<b>SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT SUM(aufh_dat_c != "0000-00-00") AS vorhanden, COUNT(*) AS gesamt FROM `k_beend_cnp`) t;</b>
Q78	<b>SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT COUNT(DATEDIFF(bek_dat, aufh_dat_c))-SUM(datediff(bek_dat, aufh_dat_c) &lt; 0) AS vorhanden, COUNT(*) AS gesamt FROM `k_beend_cnp`) t;</b>
Q79	<b>SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT COUNT(DATEDIFF(aufh_dat_c, eroef_dat_c))- SUM(datediff(aufh_dat_c, eroef_dat_c) &lt; 0) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp` E JOIN `k_beend_cnp` B ON E.id = B.kid) t;</b>
Q80	<b>SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT SUM(v_beruf_cn != "") AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t;</b>
Q81	<b>SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT COUNT(vid) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t;</b>
Q82	<b>SELECT SUM(w_k = 1)/ COUNT(*), SUM(nachlass_k = 1)/ COUNT(*), SUM(eheleute_k = 1)/ COUNT(*), SUM(gg_k = 1)/ COUNT(*), SUM(gt_k = 1)/ COUNT(*), SUM(getr_k = 1)/ COUNT(*), SUM(mj_k = 1)/ COUNT(*), SUM(gu_k = 1)/ COUNT(*), SUM(gesellter_k = 1)/ COUNT(*), SUM(abwesend_k = 1)/ COUNT(*), SUM(unbaufent_k = 1)/ COUNT(*), SUM(flucht_k = 1)/ COUNT(*), SUM(haft_k = 1)/ COUNT(*) FROM `k_eroef_cnp`;</b>
Q83	<b>SELECT SUM(ohg_k = 1)/ COUNT(*), SUM(kg_k = 1)/ COUNT(*), SUM(ag_k = 1)/ COUNT(*), SUM(kgaa_k = 1)/ COUNT(*), SUM(vvag_k = 1)/ COUNT(*), SUM(gmbh_k = 1)/ COUNT(*), SUM(eg_k = 1)/ COUNT(*), SUM(verein_k = 1)/ COUNT(*), SUM(sgs_k = 1)/ COUNT(*), SUM(liq_k = 1)/ COUNT(*) FROM `k_eroef_cnp`;</b>
Q84	<b>SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT COUNT(DATEDIFF(eroef_dat_c, t_dat))-SUM(DATEDIFF(eroef_dat_c, t_dat) &lt; 0) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t;</b>
Q85	<b>SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT SUM(ohg_k = 0 AND kg_k = 0 AND ag_k = 0 AND kgaa_k = 0 AND vvag_k = 0 AND gmbh_k = 0 AND eg_k = 0 AND verein_k = 0 AND sgs_k = 0) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t;</b>
Q86	<b>SELECT vorhanden, gesamt, vorhanden / gesamt AS quote FROM ( SELECT SUM(ohg_k = 1 OR kg_k = 1 OR ag_k = 1 OR kgaa_k = 1 OR vvag_k = 1 OR gmbh_k = 1 OR eg_k = 1 OR verein_k = 1 OR sgs_k = 1) AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t;</b>
Q87	<b>SELECT SUM(ohg_k = 1)/ COUNT(*), SUM(kg_k = 1)/ COUNT(*), SUM(ag_k = 1)/ COUNT(*), SUM(kgaa_k = 1)/ COUNT(*), SUM(vvag_k = 1)/ COUNT(*), SUM(gmbh_k = 1)/ COUNT(*), SUM(eg_k = 1)/ COUNT(*), SUM(verein_k = 1)/ COUNT(*), SUM(sgs_k = 1)/ COUNT(*) FROM ( SELECT * FROM `k_eroef_cnp` WHERE ohg_k = 1 OR kg_k = 1 OR ag_k = 1 OR kgaa_k = 1 OR vvag_k = 1 OR gmbh_k = 1 OR eg_k = 1 OR verein_k = 1 OR sgs_k = 1) t;</b>
Q88	<b>SELECT vorhanden, gesamt, vorhanden / gesamt AS quote</b>

	<pre> FROM ( SELECT SUM(gericht_cn != "") AS vorhanden, COUNT(*) AS gesamt FROM `k_eroef_cnp`) t; </pre>
Q89	<pre> SELECT AVG(datediff(bek_dat, aufh_dat)) FROM k_beend_cnp WHERE aufh_dat != "0000-00-00" AND bek_dat != "0000-00-00"; </pre>
Q90	<pre> SELECT COUNT(*) FROM `ngericht` ORDER BY `gericht` ASC; </pre>
Q91	<pre> SELECT ( SELECT COUNT(*) FROM k_beend_cnp WHERE kid IS NOT NULL AND (s_name IS NOT NULL OR s_beruf IS NOT NULL OR s_ort IS NOT NULL OR gericht IS NOT NULL OR t_dat != "0000-00-00")) / ( SELECT COUNT(*) FROM k_beend_cnp WHERE kid IS NOT NULL) AS quote, ( SELECT COUNT(*) FROM k_beend_cnp WHERE kid IS NOT NULL) AS gesamt; </pre>
Q92	<pre> SELECT COUNT(DISTINCT s_ort) FROM k_eroef_cnp WHERE gericht_cn = "München"; </pre>
Q93	<pre> SELECT e.regiogerecht_cn, e.gericht_cn, e.v_ort_cn, e.v_ort, e.s_ort, replace(e.s_ort, e.v_ort_cn, "") FROM k_eroef_cnp e WHERE e.gerichtort_cn != e.v_ort_cn AND e.v_ort_cn != "" AND e.regiogerecht_cn IN ("LG Königsberg", "LG Mannheim", "LG Straßburg") AND e.id NOT IN (35916, 25918, 35933, 10424, 49242, 49316, 49779, 52702, 50766, 52788, 43455, 31403, 30755, 50947, 50922, 2516) AND (e.v_ort_cn = e.s_ort OR replace(e.s_ort, e.v_ort_cn, "") != e.s_ort) ORDER BY e.regiogerecht_cn; </pre>
Q94	<pre> SELECT COUNT(*), E.* FROM k_eroef_cnp E WHERE E.v_beruf_cn = "Sonstiger" GROUP BY vid ORDER BY vid; </pre>
Q95	<pre> SELECT t2.* FROM ( SELECT * FROM k_eroef_cnp e WHERE eheleute_k = 0 GROUP BY s_name, gericht_cn HAVING COUNT(*) &gt; 1 ORDER BY COUNT(*) DESC, gericht_cn, s_name) t1 JOIN k_eroef_cnp t2 ON t1.s_name = t2.s_name AND t1.gericht_cn = t2.gericht_cn; </pre>
Q96	<pre> SELECT * FROM k_eroef_cnp e WHERE eheleute_k = 0 GROUP BY s_name, gericht_cn HAVING COUNT(*) &gt; 1 ORDER BY COUNT(*) DESC, gericht_cn, s_name; </pre>
Q97	<pre> SELECT * FROM k_eroef_cnp e WHERE eheleute_k = 0 GROUP BY s_name, gericht_cn HAVING COUNT(*) &gt; 2 ORDER BY COUNT(*) DESC, gericht_cn, s_name; </pre>
Q98	<pre> SELECT gericht_cn, COUNT(*) FROM ( SELECT * FROM k_eroef_cnp e WHERE eheleute_k = 0 </pre>

	<pre> GROUP BY s_name, gericht_cn HAVING COUNT(*) &gt; 1 ORDER BY COUNT(*) DESC, gericht_cn, s_name) t GROUP BY gericht_cn ORDER BY COUNT(*) DESC; </pre>
Q99	<pre> SELECT YEAR(eroef_dat_c), vid, COUNT(*) FROM k_eroef_cnp WHERE gericht_cn = "Berlin I" GROUP BY YEAR(eroef_dat_c), vid ORDER BY YEAR(eroef_dat_c), vid; </pre>
Q100	<pre> SELECT regiogericht_cn, COUNT(*)/ COUNT(DISTINCT vid) FROM k_eroef_cnp WHERE vid IS NOT NULL GROUP BY regiogericht_cn ORDER BY regiogericht_cn; </pre>

### 13. Liste der Statistiksoftware-Anweisungen (Stata)

S1	<pre> #delimit ; odbc load, exec(`" SELECT E.id AS kid, E.regiogericht_cn, E.vid, V.v_beruf_cn AS v_beruf_cnm FROM k_eroef_cnp AS E LEFT JOIN vid_cm V ON V.vid = E.vid "'); encode regiogericht_cn, generate(eregiogericht_cn); encode v_beruf_cnm, generate(ev_beruf_cnm); unique kid, by(vid) gen(vid_num); numlabel eregiogericht_cn ev_beruf_cnm, add; </pre>
S2	<pre> #delimit ; tabstat vid_num if vid != . , by(ev_beruf_cnm) statistics(count mean sd semean q p90 p95 p99 min max skewness kurtosis) longstub missing save; </pre>
S3	<pre> #delimit ; mean vid_num, over (ev_beruf_cnm) vce(cluster eregiogericht_cn); </pre>