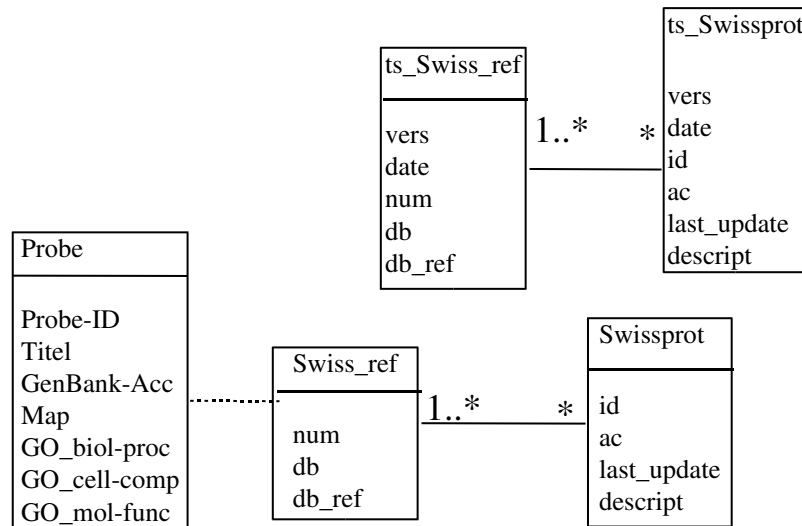


## Lösung Aufgabe4

Erweitern Sie die bestehende Datenbank um diese Informationen und zeigen Sie dies anhand eines konzeptuellen Modells mit UML und eines relationalen Modells. Laden Sie anschließend die Daten von Version 1 und Version 2 in die Datenbank. (Muß in Zukunft für n Versionen funktionieren)

10 P



### Trigger:

```

CREATE FUNCTION u_swiss() RETURNS OPAQUE AS '
DECLARE
    row RECORD;
    version INT;
BEGIN
    IF OLD.de = NEW.de AND OLD.id = NEW.id THEN
        RETURN NULL;
    ELSE
        RAISE NOTICE 'Not equal';
        SELECT max(vers) FROM ts_swissprot
        INTO row
        WHERE ac = OLD.ac;
        IF row.max IS NULL THEN
            version := -1;
        ELSE version := row.max;
        END IF;

        version := version + 1;
        INSERT INTO ts_swissprot
        VALUES (
            version,
            current_date,
            OLD.id,
            OLD.ac,
            OLD.de );
        RETURN NULL;
    END IF;
END;
' LANGUAGE 'plpgsql';

```

```

CREATE FUNCTION u_swissref() RETURNS OPAQUE AS '
DECLARE
    version INT;
    sw_vers INT;
    row RECORD;
    swiss RECORD;
    ts_swiss RECORD;
BEGIN
    IF OLD.db = NEW.db AND OLD.acc = NEW.acc THEN
        RETURN NULL;
    ELSE
        RAISE NOTICE 'Not equal reference';
        SELECT max(vers) FROM ts_swissprot_ref
        INTO row
        WHERE ac = OLD.ac
        AND num = OLD.num;
    END IF;
END;
' LANGUAGE 'plpgsql';

```

```

        IF row.max IS NULL THEN
            version := -1;
        ELSE version := row.max;
        END IF;

        version := version + 1;
        SELECT INTO swiss * FROM swissprot WHERE ac = OLD.ac;
        SELECT INTO ts_swiss * FROM ts_swissprot
            WHERE ac = OLD.ac
            AND vers = (SELECT max(vers) FROM ts_swissprot WHERE ac = OLD.ac);
    IF ts_swiss.vers IS NULL THEN
        sw_vers := -1;
    ELSE sw_vers := ts_swiss.vers;
    END IF;
        IF version = sw_vers AND current_date = ts_swiss.dat THEN
            version := version;

            ELSE
                version := sw_vers + 1;
    INSERT INTO ts_swissprot
        VALUES (
            version,
            current_date,
            swiss.id,
            swiss.ac,
            swiss.de );
    END IF;
        INSERT INTO ts_swissprot_ref
            VALUES (
                version,
                current_date,
                OLD.ac,
                OLD.num,
                OLD.db,
                OLD.acc
            );
        RETURN NULL;
    END IF;
END;
' LANGUAGE 'plpgsql';

```

```

CREATE FUNCTION d_swiss() RETURNS OPAQUE AS '
DECLARE
    row RECORD;
    version INT;
BEGIN
    SELECT max(vers) FROM ts_swissprot
        INTO row
        WHERE ac = OLD.ac;
    IF row.max IS NULL THEN
        version := -1;
    ELSE version := row.max;
    END IF;

    version := version + 1;
    INSERT INTO ts_swissprot
        VALUES (
            version,
            current_date,
            OLD.id,
            OLD.ac,
            OLD.de );
    RETURN NULL;
END;
' LANGUAGE 'plpgsql';

```

```

CREATE FUNCTION d_swissref() RETURNS OPAQUE AS '
DECLARE
    version INT;
    row RECORD;
BEGIN
    SELECT INTO row max(vers) FROM ts_swissprot_ref
        WHERE ac = OLD.ac AND num = OLD.num;
    IF row.max IS NULL THEN
        version := -1;
    ELSE version := row.max;
    END IF;
    version := version + 1;

    INSERT INTO ts_swissprot_ref
        VALUES (
            version,
            current_date,
            OLD.ac,
            OLD.num,
            OLD.db,
            OLD.acc
        );
    RETURN NULL;
END;
' LANGUAGE 'plpgsql';

```

```

CREATE TRIGGER update_swiss
AFTER update ON swissprot
FOR EACH ROW
EXECUTE PROCEDURE u_swiss();

CREATE TRIGGER delete_swiss
AFTER delete ON swissprot
FOR EACH ROW
EXECUTE PROCEDURE d_swiss();

CREATE TRIGGER update_swissref
AFTER update ON swissprot_ref
FOR EACH ROW
EXECUTE PROCEDURE u_swissref();

CREATE TRIGGER delete_swissref
AFTER delete ON swissprot_ref
FOR EACH ROW
EXECUTE PROCEDURE d_swissref();

```

Wie viele der Proben auf dem Array haben einen entsprechenden Eintrag in der aktuellen Version von SWISSPROT?

2 P

**Antwort:**

```

microarray=# select count(distinct probeid) from swissprot_ref, probe
microarray=# where db = 'EMBL'
microarray=# and acc = gbacc;
count
-----
7936
(1 row)

```

Nach wie vielen der Proteine, die in der aktuellen Version von SWISSPROT vertreten sind, können wir auf unseren gespeicherten Arrays Tests durchführen?

2 P

**Antwort:**

```

microarray=# select count(distinct ac) from swissprot_ref, probe
microarray=# where db = 'EMBL'
microarray=# and acc = gbacc;
count
-----
6237
(1 row)

```

Da ein Mitarbeiter eines Labors die Proteine in SWISSPROT nicht mit der Accession number gespeichert hat, sondern mit der ID (die sich auch ändern kann), möchte er wissen, welche ID's sich von der Alten zur Neuen Version verändert haben.

2 P

**Antwort:**

```

microarray=# select swissprot.ac, swissprot.id AS new, ts_swissprot.id AS old from swissprot,
ts_swissprot
microarray=# where swissprot.ac = ts_swissprot.ac
microarray=# and swissprot.id NOT LIKE ts_swissprot.id;
ac | new | old
---|---|---
O00141 | SGK1_HUMAN | SGK_HUMAN
O00154 | BACH_HUMAN | CTE2_HUMAN
O00409 | FXN3_HUMAN | CHE1_HUMAN
O14512 | SOC6_HUMAN | SOC7_HUMAN
O14544 | SOC4_HUMAN | SOC6_HUMAN
O14905 | WN9B_HUMAN | WN15_HUMAN
O15353 | FXN1_HUMAN | WHN_HUMAN
O15522 | NK28_HUMAN | NK2H_HUMAN
O43309 | Z305_HUMAN | Y426_HUMAN
O75110 | AT9A_HUMAN | AT2A_HUMAN
O75325 | GAMP_HUMAN | GAC1_HUMAN
O75642 | IF1H_HUMAN | IF1A_HUMAN
O75716 | STKG_HUMAN | ST16_HUMAN
O94804 | STKA_HUMAN | ST10_HUMAN
O94833 | BPEA_HUMAN | ACFX_HUMAN
O94901 | U84A_HUMAN | SUN1_HUMAN
O94956 | S219_HUMAN | OATB_HUMAN
O95096 | NK22_HUMAN | NK2B_HUMAN
O95248 | SBF1_HUMAN | MTR5_HUMAN
P04350 | TBBX_HUMAN | TBB5_HUMAN
P05218 | TBB5_HUMAN | TBBX_HUMAN
P05423 | RPC4_HUMAN | BN51_HUMAN
P08886 | RB2A_HUMAN | RAB2_HUMAN
P09086 | PO22_HUMAN | OCT2_HUMAN

```

P14859	PO21_HUMAN	OCT1_HUMAN
P20264	PO33_HUMAN	BRN1_HUMAN
P20265	PO32_HUMAN	OC3N_HUMAN
P23193	TCE1_HUMAN	TFS2_HUMAN
P27105	STOM_HUMAN	BAN7_HUMAN
P27448	MRK3_HUMAN	KP78_HUMAN
P32314	FXN2_HUMAN	HTLF_HUMAN
P35030	TRY3_HUMAN	TRY4_HUMAN
P35548	MSX2_HUMAN	HMX2_HUMAN
P36402	TCF7_HUMAN	TCF1_HUMAN
P39687	A32A_HUMAN	PHA1_HUMAN
P40222	YL14_HUMAN	LL14_HUMAN
P46721	S213_HUMAN	OATP_HUMAN
P49335	PO34_HUMAN	BRN4_HUMAN
P50748	KNTC_HUMAN	Y166_HUMAN
P50876	U7I4_HUMAN	Y161_HUMAN
P52732	KF11_HUMAN	EG5_HUMAN
P52952	NK25_HUMAN	NK2E_HUMAN
P55011	S122_HUMAN	NKC1_HUMAN
P55017	S123_HUMAN	TSCC_HUMAN
P56545	CTP2_HUMAN	CTB2_HUMAN
P56880	CLDK_HUMAN	CLDJ_HUMAN
P78367	NK32_HUMAN	HK32_HUMAN
P78412	IRX6_HUMAN	IRX7_HUMAN
P78426	NK61_HUMAN	HK61_HUMAN
P98177	FXO4_HUMAN	AFX1_HUMAN
Q01851	PO41_HUMAN	BR3A_HUMAN
Q01860	PO51_HUMAN	OC3A_HUMAN
Q02083	ASAL_HUMAN	PLT_HUMAN
Q02241	KF23_HUMAN	KNS5_HUMAN
Q02952	AK12_HUMAN	AKAC_HUMAN
Q03052	PO31_HUMAN	OCT6_HUMAN
Q06265	RR45_HUMAN	PMC1_HUMAN
Q07654	TFF3_HUMAN	ITF_HUMAN
Q12766	SMF_HUMAN	Y194_HUMAN
Q12769	N160_HUMAN	K197_HUMAN
Q12837	PO42_HUMAN	BR3B_HUMAN
Q13363	CTP1_HUMAN	CTB1_HUMAN
Q13621	S121_HUMAN	NKC2_HUMAN
Q14807	KF22_HUMAN	KNS4_HUMAN
Q14964	R39A_HUMAN	RB39_HUMAN
Q15040	JOS1_HUMAN	Y063_HUMAN
Q15319	PO43_HUMAN	BR3C_HUMAN
Q15831	STKB_HUMAN	ST11_HUMAN
Q8WXH5	SOC7_HUMAN	SOC4_HUMAN
Q92959	S212_HUMAN	PGT_HUMAN
Q93074	T230_HUMAN	Y192_HUMAN
Q96JL9	Z333_HUMAN	YI06_HUMAN
Q99546	ZWIA_HUMAN	MPP5_HUMAN
Q99547	MPH6_HUMAN	MPP6_HUMAN
Q99661	KF2C_HUMAN	MCAK_HUMAN
Q99801	NK31_HUMAN	HK31_HUMAN
Q15831	STKB_HUMAN	ST11_HUMAN
Q8WXH5	SOC7_HUMAN	SOC4_HUMAN
Q92959	S212_HUMAN	PGT_HUMAN
Q93074	T230_HUMAN	Y192_HUMAN
Q96JL9	Z333_HUMAN	YI06_HUMAN
Q99546	ZWIA_HUMAN	MPP5_HUMAN
Q99547	MPH6_HUMAN	MPP6_HUMAN
Q99661	KF2C_HUMAN	MCAK_HUMAN
Q99801	NK31_HUMAN	HK31_HUMAN
Q9BTJ7	CU97_HUMAN	YU02_HUMAN
Q9BV38	WD18_HUMAN	WDR1_HUMAN
Q9BW19	KFC1_HUMAN	KNS2_HUMAN
Q9BWJ5	S3BA_HUMAN	U186_HUMAN
Q9B XK5	BC13_HUMAN	MIL1_HUMAN
Q9BZH6	WD11_HUMAN	WDRB_HUMAN
Q9GZL7	WD12_HUMAN	WDRC_HUMAN
Q9H112	CS11_HUMAN	CT11_HUMAN
Q9H1F0	WA10_HUMAN	WAPA_HUMAN
Q9H1Z4	WD13_HUMAN	WDRD_HUMAN
Q9HBG6	WD10_HUMAN	WDRA_HUMAN
Q9HCP6	KB73_HUMAN	YB73_HUMAN
Q9NQZ8	ZN71_HUMAN	EZIT_HUMAN
Q9NUK0	MBXL_HUMAN	MBL2_HUMAN
Q9POT7	TME9_HUMAN	H186_HUMAN
Q9UBK7	RBLA_HUMAN	RB2A_HUMAN
Q9UEU0	VT1B_HUMAN	VTL1_HUMAN
Q9UH99	U84B_HUMAN	SUN2_HUMAN
Q9UIL4	KF25_HUMAN	KNS3_HUMAN
Q9UKA4	AK11_HUMAN	AKAB_HUMAN
Q9ULT0	TTC7_HUMAN	YB40_HUMAN
Q9UM07	PDI4_HUMAN	PDI5_HUMAN
Q9UNT1	RBLB_HUMAN	RB2B_HUMAN
Q9UPZ3	HPS5_HUMAN	YA17_HUMAN
Q9Y3A5	SBDS_HUMAN	YC97_HUMAN
Q9Y3B6	CNC2_HUMAN	CGX2_HUMAN
Q9Y5S9	RB8A_HUMAN	RBM8_HUMAN
Q9Y6L6	S216_HUMAN	OAT6_HUMAN

(104 rows)

An uns wird eine Anfrage gestellt, für welche Proteine, die das Wort 'apoptosis' in ihrer Description enthalten, es auch Proben auf unseren gespeicherten Arrays gibt. Geben Sie dazu die Protein und die GenBank Accession number, die ID der Probe und die dazugehörige Array ID an.

**2 P**

**Antwort:**

```
microarray=# select swissprot.ac, gbacc, probeid, "Array-Id"
microarray-# from swissprot, swissprot_ref, probe, "Array_Probe_link"
microarray-# where de ~ 'apoptosis'
microarray-# and swissprot.ac = swissprot_ref.ac
microarray-# and swissprot_ref.acc = gbacc
microarray-# and probeid = "Probe-Id";
```

ac	gbacc	probeid	Array-Id
P98170	U45880	1646_at	GPL91
P98170	U45880	1646_at	GPL74
P50591	U37518	1715_at	GPL91
P50591	U37518	1715_at	GPL74
Q13489	U45878	1717_s_at	GPL91
Q13489	U45878	1717_s_at	GPL74
O43618	AF005775	1867_at	GPL91
O43618	AF005775	1867_at	GPL74
O43618	AF005775	1868_g_at	GPL91
O43618	AF005775	1868_g_at	GPL74
O43618	AF015451	32746_at	GPL91
Q13075	U80017	33715_r_at	GPL91
O14798	AF014794	34493_at	GPL91
O14763	AF016266	34892_at	GPL91
O75414	AF051941	35005_at	GPL91
Q9UBN6	AF029761	35537_at	GPL91
Q13490	U37547	36578_at	GPL91
P56597	AF067724	36859_at	GPL91
Q9C000	AB023143	37127_at	GPL91
Q13075	U80017	37277_at	GPL91
Q13075	U80017	37313_at	GPL91
O94768	AB011421	37524_at	GPL91
O14684	AF010316	38131_at	GPL91
P55060	AF053641	38804_at	GPL91
Q9UEE5	AB011420	38822_at	GPL91
O43508	AF055872	39588_at	GPL91

(26 rows)

Bitte erstellen Sie ein dump-file ihrer Datenbank und geben sie dessen Größe an.

**2 P**