```
CREATE OR REPLACE FUNCTION check_table_a() RETURNS TRIGGER
AS $check_table_a$
  BEGIN
    IF (TG_OP = 'DELETE') THEN
      RAISE NOTICE 'deleted % recods from table b!', (SELECT count(*)
FROM b WHERE b.a_id=OLD.id);
      DELETE FROM b WHERE b.a_id=OLD.id;
      RETURN OLD;
    ELSIF (TG_OP = 'UPDATE') THEN
      IF (SELECT count(*) FROM b WHERE b.a_id=OLD.id) = 0 THEN
        RETURN NEW;
      ELSE
        RAISE EXCEPTION 'there is a row with a_id = % in table b!', OLD.id;
      END IF;
    ELSE
      RETURN NEW;
    END IF;
    RETURN NULL;
  END;
$check_table_a$ LANGUAGE 'plpgsql';

CREATE TRIGGER check_table_a AFTER INSERT OR UPDATE OR
DELETE ON a FOR EACH ROW EXECUTE PROCEDURE check_table_a();

CREATE OR REPLACE FUNCTION check_table_b() RETURNS TRIGGER
AS $check_table_b$
  BEGIN
    IF (TG_OP = 'DELETE') THEN
      RETURN OLD;
    ELSE
      IF (SELECT count(*) FROM a WHERE a.id=NEW.a_id) > 0 THEN
        RETURN NEW;
      ELSE
        RAISE EXCEPTION 'there is no row with id = % in table a!', NEW.a_id;
      END IF;
    END IF;
    RETURN NULL;
  END;
$check_table_b$ LANGUAGE 'plpgsql';

CREATE TRIGGER check_table_b AFTER INSERT OR UPDATE OR
DELETE ON b FOR EACH ROW EXECUTE PROCEDURE check_table_b();
```